ADAPTIVE RANDOM SEARCH METHODS FOR SIMULATION OPTIMIZATION

A Thesis
Presented to
The Academic Faculty

by

Andrei A. Prudius

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2007

ADAPTIVE RANDOM SEARCH METHODS FOR SIMULATION OPTIMIZATION

Approved by:

Sigrún Andradóttir, Committee Chair
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Hayriye Ayhan
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

David M. Goldsman
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Seong-Hee Kim
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Barry L. Nelson
Department of Industrial Engineering and Management Sciences
Northwestern University

Date Approved: June 5, 2007

To my family.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Dr. Sigrún Andradóttir for her guidance, support, and constant encouragement that has made this quest a success.

I would also like to thank Dr. Hayriye Ayhan, Dr. David Goldsman, Dr. Seong-Hee Kim, and Dr. Barry L. Nelson for their willingness to serve on my defense committee and for their helpful comments. I am especially grateful to Dr. Nelson for his careful proofreading and his helpful suggestions for improving my work. I am also thankful to Dr. Anton Kleywegt for serving on my proposal committee.

I also thank my friends for their help, support, and presence in my life during hard times.

Finally, I would like to thank my family for all the sacrifices they have made for me to succeed throughout my life and for their unconditional love and support. I am also thankful to my sister Irina for introducing me to mathematics at an early age.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis is concerned with identifying the best decision among a set of possible decisions in the presence of uncertainty. We are mainly interested in solving such problems in situations where the objective function value at any feasible solution cannot be evaluated exactly, but needs to be estimated, for example via a "black-box" simulation procedure. This problem is especially interesting because it addresses the optimization of the performance of complex systems that are realistically represented via simulation models, so that the problem setup is very general. Moreover, problems of this type are also of practical interest, with application areas in manufacturing, financial engineering, computer and communication systems, supply-chain management, logistics, project management, etc.

This dissertation focuses on developing adaptive random search methods for simulation optimization. The methods are adaptive in the sense that they use information gathered during previous iterations to decide how simulation effort is expended in the current iteration. We consider random search because such methods assume very little about the structure of the underlying problem, and hence can be applied to solve complex simulation optimization problems with little expertise required from an end-user. Consequently, such methods are suitable for inclusion in simulation software.

We first identify desirable features that algorithms for discrete simulation optimization need to possess in order to exhibit attractive empirical performance. Our approach emphasizes maintaining an appropriate balance between exploration, exploitation, and estimation. Exploration refers to searching globally for promising solutions within the entire feasible region, exploitation involves local search of promising subregions, and estimation refers to obtaining more precise function estimates at desirable alternatives. We also present two new and almost surely convergent random search methods that possess these desirable features. Finally, we provide numerical results that show the empirical attractiveness of our methods.

In the second part of the thesis, we develop two frameworks for designing adaptive and almost surely convergent random search methods for discrete simulation optimization. Our frameworks involve averaging, in that all decisions that require estimates of the objective function values at various feasible solutions are based on the averages of all observations collected at these solutions so far, as opposed to the averages of observations collected in the current iteration only. This feature may potentially lead to a significant reduction in the computational time required to solve the optimization problem, especially when estimating the performance measure of interest involves conducting a steady-state simulation. We also present two new and almost surely convergent variants of the simulated annealing (SA) algorithm. Finally, we provide some numerical results that demonstrate the empirical effectiveness of averaging and adaptivity in the context of SA.

Finally, we present three random search methods for simulation optimization problems with uncountable feasible regions. One of the approaches is adaptive, while the other two are based on pure random search. The only difference between the latter two approaches is the estimator of the optimal solution. The adaptive approach and one of the pure random search approaches are new to this thesis. The other pure random search approach has been proposed and analyzed before and our contribution lies in extending its convergence analysis and documenting its numerical performance. We also present conditions under which the three methods are convergent, both in probability and almost surely. Lastly, we provide a computational study that demonstrates the effectiveness of the methods when compared to some other approaches available in the literature.

CHAPTER I

INTRODUCTION

This thesis is concerned with solving the following simulation optimization problem

$$\max_{\theta \in \Theta} f(\theta) = \mathbb{E}[h_\theta(X_\theta)], \tag{1.1}$$

where $f : \Theta \to \mathbb{R}$ is the objective function, Θ is the feasible region, and for each $\theta \in \Theta$, X_θ is a random element in some space $\mathcal{X}_\theta$ and $h_\theta : \mathcal{X}_\theta \to \mathbb{R}$ is a deterministic function. We are mainly interested in solving the problem (1.1) in situations where the objective function value $f(\theta)$ at any $\theta \in \Theta$ cannot be evaluated exactly, but needs to be estimated, for example via a "black-box" simulation procedure.

It is well known that optimization via simulation is an especially difficult problem (see, e.g., Fu [32] and Banks et al. [18]). There are two main difficulties associated with this problem. In particular, it is often the case in simulation optimization that little is known about the structure of the objective function f and solving even a deterministic optimization problem with little known structure is a difficult task by itself. On top of that, the objective function value at each feasible solution of a simulation optimization problem can not be evaluated exactly, but needs to be estimated via simulation. Of course, one can argue that the second issue can be almost completely eliminated by performing a lot of simulation runs (transient simulation) or long simulation runs (steady-state simulation) at the design points to diminish the effects of the stochastic noise. However, because simulations are usually computationally expensive (so that estimating the objective function value at a single point may be computationally expensive), this would mean that only a few alternatives will be explored (see Banks et al. [18]). Hence, it is of interest to design specialized techniques to solve the problem (1.1) that will search the feasible space thoroughly and yet be able to identify optimal or nearly optimal solutions in the presence of stochastic noise.

This dissertation focuses on developing adaptive random search methods for simulation optimization. The methods are adaptive in the sense that they use information gathered

during previous iterations to adaptively decide how simulation effort is expended in the current iteration. The first part of the thesis is concerned with identifying desirable features that algorithms for discrete simulation optimization need to possess in order to exhibit attractive empirical performance. Our approach emphasizes maintaining an appropriate balance between exploration, exploitation, and estimation. With the exception of estimation, our ideas are also applicable in deterministic optimization. Exploration refers to searching globally for promising solutions within the entire feasible region Θ, exploitation involves local search of promising subregions of Θ, and estimation refers to obtaining more precise function estimates at desirable alternatives. The role of each component during various stages of the search is discussed. We also present two new random search methods that possess these desirable features. These algorithms are intuitive, simple, flexible enough to allow an end-user to exploit the structure inherent in the optimization problem of interest, and particularly suited for problems with multiple local solutions and/or steady-state simulation performance measures. We also prove their almost sure global convergence, and provide numerical results that show their empirical attractiveness.

The second part of the thesis concerns the development of two frameworks for designing adaptive and almost surely convergent random search methods for discrete simulation optimization. One of our frameworks is very broad (in that it includes many random search methods), while the other one considers a special class of random search methods, called point-based methods, that move iteratively between points within the feasible region. Our frameworks involve averaging, in that all decisions that require estimates of the objective function values at various feasible solutions are based on the averages of all observations collected at these solutions so far, as opposed to the averages of observations collected in the current iteration only. This feature may be especially useful when estimating the performance measure of interest involves conducting a steady-state simulation because the methods do not require discarding any information obtained during previous iterations of the algorithm. This potentially leads to a significant reduction in the computational time required to solve the underlying optimization problem.

We also present two new variants of the simulated annealing (SA) algorithm. SA is

a popular method among practitioners for solving both deterministic and stochastic optimization problems. However, the SA algorithms for stochastic optimization available in the literature possess the following possibly undesirable properties: (i) only the objective function observations obtained in the current iteration are used to guide the search and (ii) the number of objective function observations obtained in each iteration is often required to grow deterministically with the iteration number. One of our variants of SA does not have either drawback, while the other variant does not have drawback (ii). We show that our SA methods are almost surely convergent under mild conditions. The theoretical analysis of these algorithms yields interesting results about the behavior of the SA algorithm with decreasing cooling schedule for deterministic and stochastic optimization. For instance, we show that even though the sequence of current iterates converges to the set of global optimizers in probability, it also visits every solution infinitely often with probability one. Finally, we provide some numerical results that demonstrate the empirical effectiveness of averaging and adaptivity in the context of SA.

In the third part of the thesis, we present and analyze three random search methods for solving stochastic optimization problems with uncountable feasible regions. The three methods differ primarily in the approaches they use to reduce the effects of noise in the estimated objective function values. The first method achieves this goal through the occasional resampling of already sampled points, while the other two approaches address it by averaging observations in balls that shrink with time (one of these methods was originally proposed and analyzed by Baumert and Smith [20]). The methods also differ in that the first approach is adaptive (in that certain algorithmic decisions can be based on all the information collected by the method so far), and may consequently involve local search. The other two approaches are based on pure random search, with the only difference being the estimator of the optimal solution. We present conditions under which the three methods are convergent, both in probability and almost surely. Finally, we provide a computational study that demonstrates the effectiveness of the three methods when compared to some other random search approaches available in the literature. In particular, our first (adaptive) approach exhibits overall good empirical performance, especially on problems

for which the probability of identifying "good" solutions using pure random search is small.

The remainder of this thesis is organized as follows. In Chapter 2 we present a short literature review. In Chapter 3 we discuss desirable features that optimization algorithms need to possess to exhibit good empirical performance when applied to solve deterministic or simulation optimization problems having little known structure. We also present two new random search methods that possess these desirable properties, prove their almost sure global convergence, and provide numerical results for the proposed methods that show their attractive empirical behavior. In Chapter 4 we propose two frameworks based on averaging for designing random search methods for discrete simulation optimization, present two new variants of the SA algorithm and discuss their convergence properties, and demonstrate the empirical effectiveness of averaging and adaptivity in the context of SA. In Chapter 5 we present and analyze three random search methods for solving stochastic optimization problems with uncountable feasible regions, prove their convergence, and show empirically their attractiveness when compared to some other random search approaches available in the literature. Our conclusions and future research directions are given in Chapter 6.

CHAPTER II

LITERATURE REVIEW

Simulation optimization is concerned with identifying optimal design parameters for a stochastic system, where optimal is measured by an expectation of a function of output variables associated with a simulation model. This topic has received tremendous attention from the research community in the past two decades. Simulation optimization also has been increasingly applied in practice. Application areas include manufacturing (e.g., Morito et al. [64] and Vogt [89]), supply-chain management (e.g., Fu and Healy [34], Azadivar, Shu, and Ahmad [17], and Truong and Azadivar [87]), logistics (e.g., Hill and Fu [45] and Wieland and Holden [90]), and project management (e.g., April et al. [16]).

A number of excellent surveys on the topic have been written. For example, Carson and Maria [27] provide a general review of simulation optimization, including discussion on solution approaches, applications, and commercially available software. Swisher et al. [86] present another survey, focusing on gradient and non-gradient approaches for continuous input parameters, statistical methods for problems with a small number of feasible solutions, and random search and ordinal optimization methods for problems with a large number of feasible solutions. Andradóttir [10] also presents a review on simulation optimization techniques, mainly focusing on gradient estimation, stochastic approximation, and random search methods. Goldsman and Nelson [39] offer a survey of ranking and selection and multiple comparison procedures, while Kim and Nelson [56] explain how such procedures are constructed and review key results that are useful in designing such procedures. Fu [32] provides a tutorial that summarizes solution approaches, discusses implemented algorithms in commercial software, and comments on promising research areas and possible future directions. Andradóttir [13] provides an overview of random search methods, discusses their convergence, and describes desirable features that random search methods need to have to exhibit attractive empirical performance.

The above review of survey papers on simulation optimization is by no means exhaustive. In this chapter we provide a brief literature review, mainly focusing on random search methods. For more detailed overviews on the topic, the interested reader is referred to the aforementioned manuscripts and references therein.

Most of the existing research on solving the problem (1.1) can be further subdivided into settings when the feasible region Θ is either discrete or continuous. In Section 2.1 we discuss methods designed for solving discrete simulation optimization methods, while in Section 2.2 we present a review of methods devised for solving continuous simulation optimization problems.

2.1 Discrete Decision Parameters

In this section we discuss methods designed for solving discrete simulation optimization problems, mainly focusing on random search methods. Random search methods usually assume very little about the topology of the underlying optimization problem and about the objective function observations, and only require estimates of the objective function values to be available.

Yan and Mukai [95] propose the stochastic ruler method. This is an iterative method that moves from a (current) feasible point to another candidate solution based on an objective function estimate at the candidate solution and the value of a uniform random variable called the stochastic ruler. This approach is globally convergent in probability. Alrefaei and Andradóttir [4, 5] present modifications of the stochastic ruler method and show that their variants are almost surely convergent and numerically more efficient than the original method.

Andradóttir [6, 8] develops stochastic comparison methods that differ from the stochastic ruler method in that the comparison is carried out with an estimate of the objective function value at the current solution rather than a stochastic ruler. The method presented in Andradóttir [6] is locally convergent with probability one even when the feasible region is countably infinite, while the method in Andradóttir [8] is globally convergent with probability one. Gong, Ho, and Zhai [40] also propose a stochastic comparison method,

prove its global convergence in probability, and compare it numerically to the stochastic ruler method of Yan and Mukai [95]. Andradóttir [11] presents a variant of the stochastic comparison method of Gong, Ho, and Zhai [40], proves its almost sure global convergence, and discusses its convergence rate. Unlike the method proposed by Gong, Ho, and Zhai [40], the methods of Andradóttir [6, 8, 11] do not require collecting an increasing number of objective function observations per iteration as the number of iterations grows.

In the context of random search methods, Andradóttir [11] proposes to use the solution with the highest estimated objective function value as the estimate of the optimal solution. She also discusses advantages of this approach and presents several rate of convergence results for random search methods using the above approach to estimate the optimal solution. When the feasible region is countably infinite, Andradóttir [14] suggests using the solution with the highest estimated objective function value as the estimate of the optimal solution, provided that this solution has been visited "often enough." She also presents a class of random search methods for simulation optimization with countably infinite regions and analyzes their convergence.

The simulated annealing (SA) algorithm originally proposed for deterministic optimization also has been applied to discrete simulation optimization. Gelfand and Mitter [36] present the convergence analysis of a SA algorithm applied to solve the problem (1.1). They show that if errors in the objective function estimates are normally distributed with mean zero and variance decreasing asymptotically faster than the cooling schedule, then the method is convergent in probability. Gutjahr and Pflug [41] also analyze the method of Gelfand and Mitter [36] and show that it converges in probability provided that the variance of normally distributed errors decreases at a rate that is significantly faster than the cooling schedule. Moreover, they extend their analysis to errors that are more peaked around zero than normally distributed errors. Fox and Heine [31] also develop a variant of SA that is convergent in probability. They do not make restrictive assumptions about the distribution of errors in the objective function estimates, but they assume that these estimates are restricted to a finite set. Also, the objective function estimate at any given feasible point is the average of all simulation observations collected at this point so far.

Alrefaei and Andradóttir [3] present two variants of SA with a constant cooling schedule and demonstrate their almost sure convergence and numerical efficiency with respect to the other SA algorithms for stochastic optimization reviewed above.

Shi and Ólafsson [83] propose the nested partitions (NP) method for discrete simulation optimization. At any given point in time, this method focuses the search on the current "most promising" subregion of the feasible space. In particular, each step of the method involves partitioning the current most promising region, random sampling of solutions within the most promising and surrounding regions, updating the most promising region, and backtracking if a better solution is found outside the current most promising region. They show that the approach is almost surely convergent. Shi and Ólafsson [84] show that the Markov chain of most promising regions generated by the NP method converges to the stationary distribution and use these results to derive a stopping criterion for the method. Pichitlamken and Nelson [71] enhance the numerical performance of the method by changing the estimator of the optimal solution and incorporating a statistical procedure and local improvement schemes into the approach. They show that their variant is also almost surely convergent.

Hong and Nelson [51] propose the COMPASS method for discrete simulation optimization (with each feasible solution being an integer-valued vector). This method focuses its sampling effort in the current most promising region, which consists of all feasible points that are closer (according to Euclidean distance) to the "best" solution found so far (i.e., the solution with the highest estimated objective function value) than to any other sampled point. This method is almost surely convergent to a locally optimal solution (i.e., a solution with a higher objective function value than all solutions with Manhattan distance one from this solution) in contrast to most of the random search methods discussed before, which are globally convergent. More recently, Hong [52] presents the coordinate search method (which is not a random search method) that is also almost surely locally convergent, and compares its performance to COMPASS.

Ho, Srinivas, and Vakili [48] propose a paradigm known as ordinal optimization for solving discrete simulation optimization problems. This approach is especially effective

on problems with finite but large feasible regions. The main idea of their methodology is to soften the goal when solving the problem (1.1) from finding an optimal solution to identifying a "good enough" solution (this significantly reduces the computational burden). Then they suggest sampling a number of alternatives from the feasible region using pure random search, conducting simulations at these alternatives to obtain a rough ranking of these alternatives, and then discard all but the top r designs, where r is significantly smaller than the cardinality of Θ. Finally, they propose applying any discrete simulation optimization algorithm to identify good designs among the remaining r solutions. Although there is a risk in retaining only r solutions out of the entire feasible region in that these points might not contain good solutions, Ho, Srinivas, and Vakili [48] show that the probability of this occurring is often very low. Additional references on the topic include Dai [29], Dai and Chen [30], and Ho [46].

A number of statistically valid ranking-and-selection (R&S) procedures have been developed to solve the problem (1.1) when the number of simulated systems is finite. The common feature of these methods is that at their termination they provide a statistical guarantee regarding the quality of the chosen system. Recent references include Nelson et al. [66], Kim and Nelson [54, 55], and Batur and Kim [19]. Although the traditional role of R&S methods is to select the best system from among a small number of simulation alternatives, recently R&S procedures have been designed for comparing a larger number of systems. R&S methods can be embedded in random search methods designed for simulation optimization. For example, Boesel, Nelson, and Kim [25] suggest applying their R&S procedure to identify the best design from among "good" alternatives identified by a random search method, while Pichitlamken and Nelson [71] use their procedure to aid their random search algorithm to make better moves more efficiently. More recently, Andradóttir and Kim [15] develop R&S procedures that identify the best system among a finite number of alternatives in the presence of a stochastic constraint on a secondary measure of interest. Excellent surveys on R&S methods can be found in Goldsman and Nelson [39] and Kim and Nelson [56].

2.2 Continuous Decision Parameters

In this section we provide a short overview of methods designed for solving continuous simulation optimization problems. In particular, most of the existing research aimed at solving the problem (1.1) when Θ is uncountable involves estimating the gradient (and possibly higher order derivatives) of the objective function f. Stochastic approximation is one popular class of such methods. The first stochastic approximation algorithm was proposed by Robbins and Monro [75]. When applied to solve the problem (1.1), this algorithm is essentially a generalization of the steepest descent method of deterministic optimization for the context of stochastic optimization. In particular, in each iteration k this method moves from a point θ_k to another point $\theta_{k+1} = \theta_k + a_k \cdot \nabla \hat{f}(\theta_k)$, where $\nabla \hat{f}(\theta)$ is an estimate of the gradient of f at θ and $\{a_k\}$ is a sequence of positive numbers decreasing to zero. A lot of effort has been expended on understanding the practical and theoretical aspects of stochastic approximation methods. Some work on this topic includes books by Kushner and Clark [58], Benveniste, Métivier, and Priouret [21], Ljung, Pflug, and Walk [61], and Kushner and Yin [59] and articles by Ruppert [79], Polyak and Juditsky [72], Andradóttir [7, 9], Bhatnagar and Borkar [22], and L'Ecuyer and Yin [60].

Another class of methods that has been developed to solve continuous simulation optimization problem is known as the sample average approximation (SAA) method, stochastic counterpart approach, or retrospective optimization. The basic idea of these methods is to generate a random sample, approximate the expected value function f by a corresponding sample average function, and finally use standard mathematical programming techniques to locate the optimal solution of this deterministic function. One advantage of these approaches is that they also allow constraints to be stochastic (and not just the objective function) because these constraints can again be approximated via the corresponding sample average functions, and then techniques for constrained optimization can be used to solve the resulting deterministic problem. Some work on SAA methods includes Healy and Schruben [44], Robinson [76], Shapiro [80], Shapiro and Wardi [81], Mak, Morton, and Wood [62], Kleywegt, Shapiro, and Homem-de-Mello [57], and Blomvall and Shapiro [23].

One crucial component in applying stochastic approximation and SAA methods to solve

the optimization problem (1.1) is the estimation of the gradient of the objective function (and possibly some higher order derivatives). This can be accomplished via general procedures like finite (forward or central) differences and simultaneous perturbations (see, for instance, Spall [85]), or with more specialized techniques like the infinitesimal perturbation analysis, the likelihood ratio method, weak derivatives (see, for instance, Pflug [69, 70]), etc. These specialized techniques are usually more efficient from a computational point of view, but require special problem structure and expertise from the end-user. The literature on gradient estimation is vast and some work on this topic includes books by Glasserman [38], Ho and Cao [47], Rubinstein and Shapiro [78], and Fu and Hu [35], and a recent overview article by Fu [33].

In the past, a few random search methods have been proposed for solving continuous simulation optimization problems. In particular, Yakowitz and Lugosi [94] develop a method that at certain iterations samples new solutions from a fixed global distribution and ensures that every sampled point has "enough" observations, and at other times it adaptively resamples previously sampled points. Yakowitz, L'Ecuyer, and Vázquez-Abad [93] propose two approaches that utilize low-dispersion point sets and emphasize how the number of such points should be determined depending on the problem and the simulation budget. Baumert and Smith [20] propose an approach based on pure random search that estimates the objective function value at each solution θ by averaging all observations within a certain distance from θ, and discuss how this distance should decrease in order for the method to converge in probability. Alexander et al. [2] develop a procedure that is convergent in probability and that iteratively samples a solution from Θ based on a fixed sampling strategy and then compares the incumbent and sampled solutions using increasingly precise (as the number of iterations grows) estimates of the objective function values at these solutions. Ghate and Smith [37] study a generalized simulated annealing procedure that is similarly convergent in probability and involves comparing increasingly precise estimated objective function values as the number of iteration grows. Finally, Yakowitz [92] presents a method that combines random search with stochastic approximation, Rubinstein and Kroese [77] discuss the use of the cross-entropy method for optimizing noisy objective functions, and

Hu, Fu, and Marcus [53] present a stochastic model reference adaptive search (SMRAS) method for stochastic optimization. Both the cross-entropy and SMRAS methods can be used for solving the problem (1.1) with discrete and continuous decision parameters.

CHAPTER III

BALANCED EXPLORATIVE AND EXPLOITATIVE SEARCH WITH ESTIMATION

3.1 Introduction

In this chapter, we propose a general framework for simulation optimization, called Balanced Explorative and Exploitative Search with Estimation (BEESE). With the exception of estimation, our ideas also apply to the deterministic setting, where the framework will be referred to as BEES. More specifically, we discuss how simulation optimization algorithms should maintain an appropriate balance between exploration, exploitation, and estimation to show good numerical performance. Here exploration refers to searching globally for promising solutions within the entire feasible region Θ, exploitation involves local search of promising subregions of Θ, and estimation refers to obtaining more precise function estimates at desirable alternatives and an improved estimator of the optimal solution. The role of these three algorithm components during various stages of the search is discussed.

The ideas of exploration and exploitation (or diversification and intensification) have been used extensively in the literature on Tabu Search, Genetic Algorithms, and Nested Partitions (see, e.g., Pichitlamken and Nelson [71]). However, to the best of our knowledge, the idea of explicit incorporation of an estimation component into random search methods and the importance of doing so for achieving good numerical performance (both from the perspective of guiding the search and estimating the optimal solution) have never been discussed in the literature so far. Other authors have recently proposed random search methods that incorporate statistical procedures to ensure valid selection in each iteration of the algorithm (see, e.g., Ahmed and Alkhamis [1] and Pichitlamken and Nelson [71]). Although these statistical procedures can be viewed as estimation components of the resulting optimization methods, this is not pointed out by the original authors who use these procedures to guide the search. Moreover, our approach to estimation does not necessarily

entail incorporation of statistical procedures into the method (see Sections 3.2 and 3.5.3 for details).

We also develop two new and almost surely convergent random search algorithms, called Randomized Balanced Explorative and Exploitative Search (with Estimation) and abbreviated as R-BEES(E), and Adaptive Balanced Explorative and Exploitative Search (with Estimation) and abbreviated as A-BEES(E). Our random search methods are relatively simple, general enough to allow the end-user to take advantage of structure present in the problem by using local sampling distributions, and also have good empirical performance.

Although we are interested in solving optimization problems with little known structure, we will make a structural assumption about the objective function f. The reason is that No Free Lunch Theorems for deterministic optimization (see Wolpert and Macready [91]) show that the average performance of each algorithm over all possible discrete optimization problems is identical. This suggests that an optimization problem will only be solved efficiently if it possesses some known structure and the optimization algorithm exploits that structure. Consequently, we assume that solutions located close to each other have similar performance, which is usually the case in simulation optimization. This assumption is made implicitly by other algorithm developers and is only required to ensure that the proposed methods are numerically efficient; it is not required for proving convergence.

The remainder of this chapter is organized as follows. In Section 3.2, we discuss the features that simulation optimization algorithms should have to be efficient in practice. In Section 3.3, we present the R-BEES and R-BEESE methods for solving deterministic and stochastic optimization problems, respectively, and analyze their convergence properties. In Section 3.4, we develop the A-BEES and A-BEESE algorithms for deterministic and stochastic optimization, respectively, and provide the associated convergence analyses. Section 3.5 contains numerical examples that support the ideas discussed in Section 3.2 and illustrate the numerical performance of the newly proposed methods. Concluding remarks are given in Section 3.6. A preliminary version of this chapter appeared in Prudius and Andradóttir [73].

3.2 Framework

In this section, we discuss desirable properties that optimization algorithms should possess in order to be efficient numerically when applied to solve optimization problems with little known structure. Our BEES framework for deterministic optimization involves maintaining balance between exploration and exploitation, while our BEESE framework for simulation optimization maintains balance between exploration, exploitation, and estimation.

First we argue that it is important to maintain balance between exploration and exploitation during the search for an optimal solution. Suppose that one is interested in solving a deterministic optimization problem (i.e., $f(\theta)$ in (1.1) can be calculated without noise for every $\theta \in \Theta$) with little information about the structure of the objective function (we only know that solutions located close to each other have similar performance). Then it would be reasonable to start the search by exploring the entire feasible region (global search) to assess how the objective function behaves over the feasible space (because if little is known about the objective function, then it is likely that the search is started far from the optimal solution(s) and hence it might take a long time to identify a good subregion of the feasible space using local search). But once a good subregion is identified, the search should exploit it by searching locally for better solutions (because the probability of identifying better solutions using global search decreases as the search proceeds). Note that exploitation can be done in several promising regions simultaneously. The above discussion suggests that the effectiveness of the search algorithm depends heavily on the ability of the method to identify when it should switch focus from global search (exploration) to local search (exploitation).

Unfortunately, since little is known about the structure of the underlying problem, it is difficult to identify when to switch from exploration to exploitation. Observe that if this switch is performed too early, then a non-optimal subregion is locally searched, and if it is performed too late, then too much effort is expended on locating a good subregion. In both cases the convergence to the optimal solution(s) might be slow. In the experience of the authors, the identification of an appropriate switch point can be almost as difficult as solving the original problem (1.1). This point is illustrated in Figure 3.1 that shows typical sample

paths of $f(\theta_n^*)$ as a function of n, where θ_n^* is the point with the best objective function value found in the first n iterations of an optimization algorithm (the data in Figure 3.1 was obtained from the two hills problem with no noise described in Section 3.5.1 below). More specifically, this figure shows five sample paths of an optimization algorithm with the switch performed optimally, too early, too late (after 86, 50, and 250 objective function evaluations, respectively), or not at all (so that only exploration or exploitation is done). Observe that the convergence to the optimal solution is much slower when the switch is performed suboptimally. Moreover, the optimal solution is not found in the first 500 iterations when the switch is performed too early or not at all. Given the difficulty in determining when to switch from exploration to exploitation, our approach takes a different perspective. In particular, we propose maintaining an appropriate balance between exploration and exploitation during various stages of the search, rather than switching the focus from exploration to exploitation.

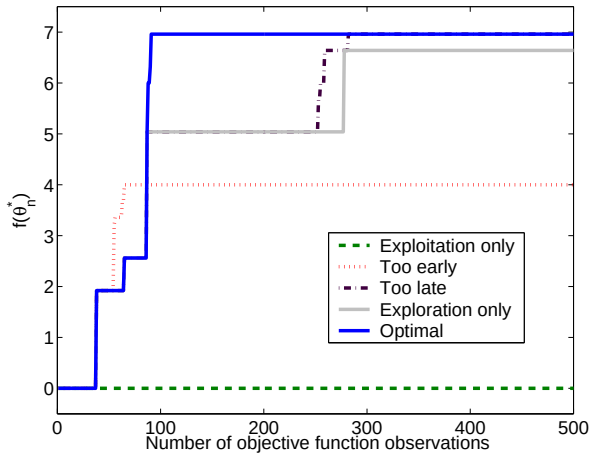


Figure 3.1: Identification of a proper switch point from exploration to exploitation

The primary difference between simulation optimization and deterministic optimization is the presence of stochastic errors in the estimated objective function values. Potentially this leads to two complications, namely that it is more difficult to effectively guide the search for improved solutions, and also to select the best solution identified by the search. Another difference between simulation optimization and deterministic optimization is that the objective functions in simulation optimization problems are more likely to possess little known structure.

Simulation optimization algorithms generally decide where future simulations are to be conducted based on the function value estimates at the points where simulations have been conducted previously. Observe though that it is possible that some alternatives might appear to be good (have high estimated objective function values) while in fact they are bad and vice versa. Hence, it is important that the optimization method not be mislead by such information for long with respect to identifying good solution(s) (see also Andradóttir [13]). This suggests that it is imperative to strategically consider where additional simulations should be conducted in order to benefit the search the most (e.g., by obtaining improved function estimates at the points with the best function estimates) and to be careful in deciding how much weight to put on the available function estimates, especially in choosing the estimate of the optimal solution. This issue will be further referred to as estimation.

In simulation optimization settings, the objective function value difference between optimal and good solutions is often small compared to the standard deviations of the objective function estimates. Thus, obtaining more precise function estimates becomes crucial toward the end of the search when the main issue is identifying an optimal solution among very good solutions, rather than on locating good alternatives. This increased emphasis on estimation can, for instance, be achieved by searching desirable regions locally (given that solutions located close to one another have similar performance) or by allocating simulation effort to points with good (high, see (1.1)) estimated objective function values. Consequently, there are settings in which local search may be desirable in stochastic optimization but not in deterministic optimization (e.g., if the objective function values at all alternatives within a desirable region already have been evaluated) and allocating additional effort to points with high estimated objective function values is never desirable for deterministic optimization. The discussion above shows that for a simulation optimization algorithm to have good empirical performance, it is important to incorporate features into the method that aid estimation. This explains why our approach to simulation optimization involves maintaining an appropriate balance between exploration, exploitation, and estimation.

Strategic estimation is usually not incorporated explicitly in simulation optimization methods (especially when deterministic optimization methods are adapted for stochastic

optimization). This is a missed opportunity from the perspective of obtaining attractive numerical performance. Next we give two examples of random search methods in which estimation is not incorporated explicitly. We also argue that the good empirical performance of these methods is at least to some extent due to the fact that they happen to do estimation well (this observation has not been made by the original authors).

First, consider a variant of the Simulated Annealing (SA) algorithm with constant temperature (Algorithm 2 in Alrefaei and Andradóttir [3]). In every iteration of the method, a candidate solution is sampled from the neighborhood of the current solution. Then simulations are conducted at the current and candidate solutions and these samples are used to make a probabilistic decision (that depends also on the temperature) on the next iterate (i.e., the current solution in the next iteration). Using samples obtained only in the current iteration allows the method to preserve a Markovian property, which is used to prove the convergence of the method. Observe that in all except the first iteration of the method, some number of simulations have already been performed at the current iterate, i.e., some estimation has taken place. Although these previously obtained simulation results are not used to guide the search, they are accumulated and used to select the estimate of the optimal solution, and consequently their use improves the empirical performance of the method. Moreover, it has been shown both empirically and theoretically that the method is attracted to good points and hence more precise estimates are quickly obtained at the good points.

As another example, consider the Nested Partitions (NP) method of Pichitlamken and Nelson [71]. The NP method has both diversification (exploration of the surrounding region) and intensification elements. The intensification component involves sampling the most promising region, applying a statistical ranking-and-selection procedure called Sequential Selection with Memory (SSM) to the sampled points, and running a hill-climbing algorithm starting from the point with the best estimated objective function value. In the initial stages of the search, intensification can be viewed as exploitation or local search but as the method progresses (i.e., the promising region becomes small and contains an optimal solution), it mainly serves estimation purposes. In essence, the method samples desirable points and the SSM procedure expends simulation effort on obtaining improved function

estimates at these points. This feature enables the method to identify the optimal solution efficiently at the end of the search and hence have good empirical performance.

Another important consideration in the design of simulation optimization algorithms is the estimation of the optimal solution. This issue is easy to resolve in the deterministic optimization setting where one can let the solution with the highest objective function value found so far be the estimate of the optimal solution, but is more challenging in the simulation optimization setting due to the noise in the estimated objective function values. The most commonly considered estimates of the optimal solution in stochastic optimization are the current solution, the most visited solution, the solution with the best estimated objective function value, and the solution with the best estimated objective function value provided it has been simulated "sufficiently often." A more thorough discussion on this issue can be found in Andradóttir [11, 13, 14]. Here, we consider the estimate proposed in Andradóttir [14], namely it is a solution with the highest estimated objective function value among solutions that have been sampled sufficiently often. We show that even though this approach for estimating the optimal solution was originally proposed for simulation optimization problems with countably infinite feasible spaces, it nevertheless exhibits good empirical performance when the feasible space is finite.

The BEES/BEESE framework presented in this section is general enough to include most random search methods available in the literature in the sense that these methods can be decomposed into our three building components (i.e., exploration, exploitation, and estimation). Hence, it is not surprising that these random search methods often perform well in practice. Our framework is also general enough to include pure random search as a special case. In the next two sections, we propose and analyze two new random search methods that possess the desired features discussed in this section.

In the next two sections, we propose two new random search methods that possess the desired features discussed in this section. In particular, the R-BEES and R-BEESE methods for deterministic and stochastic optimization, respectively, and the associated convergence analyses are given in Section 3.3, while in Section 3.4, we present the A-BEES and A-BEESE methods for deterministic and stochastic optimization, respectively, and provide

their convergence analyses.

3.3 The Randomized BEES and BEESE methods

3.3.1 Deterministic Optimization Using R-BEES

The R-BEES method for deterministic optimization randomly samples new alternatives from two families of sampling distributions. The global sampling distribution G (the only distribution in the first family) is designed for searching the entire feasible region (exploration), while the family of local sampling distributions $\mathcal{L}$ aims at searching promising subregions (exploitation). At any iteration, with probability $0 < p \leq 1$ the global sampling distribution is used and with probability $1 - p$ a local sampling distribution in $\mathcal{L}$ is used. This creates a balance in the use of exploration and exploitation during all stages of the search. Note that the R-BEES method is equivalent to using local distributions only with $L(A) = pG(A) + (1 - p)L'(A)$ for all $A \subset \Theta$, where $L' \in \mathcal{L}$. In this sense our framework includes local search as a special case. The pseudo-code for the R-BEES method is given in Algorithm 3.1.

Algorithm 3.1 (R-BEES Algorithm)

$n \leftarrow 0$
Sample a solution θ from the global distribution G
Evaluate the objective function at θ
$\theta_n \leftarrow \theta$
while Stopping criterion is not satisfied **do**
  Draw a uniform $(0, 1)$ random variable U independent of everything else
  if $U \leq p$ **then**
    Sample a solution θ from the global distribution G independent of everything else
  else
    Sample a solution θ from a local distribution in $\mathcal{L}$
  end if
  Evaluate the objective function at θ (if needed)
  $n \leftarrow n + 1$
  if $f(\theta) > f(\theta_n)$ **then**
    $\theta_n \leftarrow \theta$
  end if
end while
Present $\theta_n^* = \theta_n$ as the estimate of the optimal solution

First, a few comments about the R-BEES method are in order. Observe that it is not necessary to evaluate the objective function at a sampled solution θ if this solution has been

sampled previously. Also, in all methods proposed in this chapter, a local sampling distribution can be chosen adaptively from $\mathcal{L}$ based on the information gathered by the search method without affecting its convergence guarantee. For example, it might be desirable to focus local search around points that have high objective function values (for example, θ_n). Finally, note that the R-BEES method includes pure random search as a special case (take $p = 1$).

Suppose that all random elements in the R-BEES method (ones needed for sampling solutions) are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and let $f^* = \sup_{\theta \in \Theta} f(\theta)$. Our convergence result for the R-BEES method is given in Theorem 3.1. Observe that Theorem 3.1 is very general in the sense that it covers settings where Θ is uncountable.

Theorem 3.1. *(i) Suppose that $f^* < \infty$. Assume that the global sampling distribution G on Θ is such that for every $k \in \mathbb{N} \setminus \{0\}$, $G(A_k) > 0$, where $A_k = \{\theta \in \Theta : f(\theta) \geq f^* - 1/k\}$. Then with probability one, $f(\theta_n^*)$ converges to f^* as $n \to \infty$.*

(ii) Suppose that $f^* = \infty$. Assume that the global sampling distribution G on Θ is such that for every k integer, $G(B_k) > 0$, where $B_k = \{\theta \in \Theta : f(\theta) \geq k\}$. Then with probability one, $f(\theta_n^*)$ diverges to $+\infty$ as $n \to \infty$.

Proof: (i) For $k \in \mathbb{N} \setminus \{0\}$ and $n \in \mathbb{N}$, define Ω_k^n to be the subset of Ω such that the R-BEES method samples a solution in the set A_k at iteration n using the global sampling distribution G. Fix $k \in \mathbb{N} \setminus \{0\}$ and let $\Omega_k = \{\omega \in \Omega : \Omega_k^n \text{ i.o.}\}$ (i.o. stands for infinitely often). Observe that $\mathbb{P}(\Omega_k^n) = pG(A_k) > 0$. Then $\sum_{n=0}^{\infty} \mathbb{P}(\Omega_k^n) = \sum_{n=0}^{\infty} pG(A_k) = \infty$. Note that $\{\Omega_k^n\}_{n=0}^{\infty}$ are independent events. Then the second Borel-Cantelli lemma yields that $\mathbb{P}(\Omega_k) = 1$. Let $\tilde{\Omega} = \bigcap_{k=1}^{\infty} \Omega_k$. Obviously $\mathbb{P}(\tilde{\Omega}) = 1$. Fix $\omega \in \tilde{\Omega}$ and $l \in \mathbb{N} \setminus \{0\}$. Then there exists an iteration number $N_l(\omega)$ such that some solution in the set A_l is sampled at iteration N_l. As l is arbitrary, we get that $f(\theta_n^*) \to f^*$ as $n \to \infty$ for this ω. This concludes the proof of (i).

(ii) The proof is the same as in (i) except that A_k needs to be substituted by B_k. ■

Remark 3.1. Observe that the R-BEES method searches the feasible region using both fixed and adaptive components (i.e., the global sampling distribution and the family of

local sampling distributions, respectively). Also, it should be obvious that the conditions of Theorem 3.1 are far from necessary. For example, the R-BEESE method can be made highly adaptive without losing its convergence guarantee by letting the probability of using a global sampling distribution (p_n) and the global sampling distribution (G_n) depend on the iteration number n and the sample path of the method up to iteration $n-1$ (i.e., p_n and G_n are adapted). The convergence results given in Theorem 3.1 still apply to this variant provided that for all $k \in \mathbb{N} \setminus \{0\}$, $\sum_{n=1}^{\infty} p_n G_n(A_k) = \infty$ with probability one in (i) and $\sum_{n=1}^{\infty} p_n G_n(B_k) = \infty$ with probability one in (ii). The proof is based on the conditional Borel-Cantelli lemma (see page 32 in Hall and Heyde [43]). This and the fact that no structural assumptions on the family of local sampling distributions are made in Theorem 3.1 (e.g., the local sampling distributions need not be local in the sense of Section 3.2) imply that algorithm parameters are allowed to vary based on the previous history without affecting the convergence guarantee.

3.3.2 Stochastic Optimization Using the R-BEESE Method

The R-BEESE method for stochastic optimization is essentially the same as the R-BEES method for deterministic optimization with the addition of an estimation component. In particular, to facilitate estimation, the sampling distribution specified in lines 6 through 11 of Algorithm 3.1 is modified as follows: with probability $0 \leq \alpha < 1$ (independently of everything else), the point θ_n with the highest estimated objective function value is sampled and with probability $1-\alpha$ the sampling is done as before. Moreover, in the case of transient simulation, whenever a solution is sampled, m simulation replications are conducted at it.

We also modify the last step of the algorithm. To estimate the optimal solution we use the estimator proposed in Andradóttir [14]. More specifically, $\theta_n^* \in \Theta$ is chosen to be the estimate of the optimal solution in iteration n if it has the highest estimated objective function value among solutions that have been simulated at least M_n times. If the set of systems that have been simulated at least M_n times is empty, then the estimate of the optimal solution is the solution θ_n with the highest estimated objective function value, regardless of how many simulations have been conducted at this point. In case of deterministic optimization,

note that R-BEESE with $\alpha = 0$, $M_n = 1$, and $m = 1$ reduces to the R-BEES method.

Next we analyze the convergence properties of the R-BEESE method when applied to solve a stochastic optimization problem. We first give a few definitions. For each $\theta \in \Theta$, define $f_n(\theta)$ to be the estimate of the objective function value $f(\theta)$ available at the end of iteration n (let $f_n(\theta) = -\infty$ if $C_n(\theta) = 0$, where $C_n(\theta)$ is the number of times an alternative θ has been simulated by the end of iteration n) and $\hat{f}_k(\theta)$ to be the estimate of $f(\theta)$ after θ has been sampled k times. We need the following technical assumption.

Assumption 3.1. *For each $\theta \in \Theta$,*

$$\mathbb{P}\left\{\lim_{k\to\infty} \hat{f}_k(\theta) = f(\theta)\right\} = 1.$$

Assumption 3.1 can be easily satisfied in practice. In the case of transient simulation, let X_θ^i be the i^{th} observation of X_θ collected by the R-BEESE algorithm. Then Assumption 3.1 holds with $\hat{f}_k(\theta) = \sum_{i=1}^{km} h_\theta(X_\theta^i)/km$, provided that the sequence $X_\theta^1, X_\theta^2, \ldots$ are independent random elements with the distribution of X_θ and $\mathbb{E}[|h_\theta(X_\theta)|] < \infty$ (this follows from the Strong Law of Large Numbers).

In the case of steady-state simulation, Assumption 3.1 also can be satisfied. Below we present two ways of doing so. Usually in the steady-state simulation setting,

$$f(\theta) = \lim_{t\to\infty} \frac{1}{t} \int_0^t h'_\theta(X_\theta(u))\,\mathrm{d}u \tag{3.1}$$

almost surely, where h'_θ is a deterministic function and X_θ is a continuous time stochastic process $\{X_\theta(t) : t \geq 0\}$. For conditions that guarantee that the limit in equation (3.1) exists and equals a constant almost surely, see for example Theorem 2.2 in Chapter 2 of Shedler [82]. Hence, Assumption 3.1 can be satisfied by letting $\hat{f}_k(\theta) = \int_0^{t_k} h_\theta(X_\theta(u))\,\mathrm{d}u/t_k$, where $\{t_k\}_{k=1}^\infty$ is a sequence of positive numbers such that $t_k \to \infty$ as $k \to \infty$.

Suppose now that $\{t_k\}_{k=1}^\infty$ is nondecreasing and let $t_0 = 0$. The implication of the result above for steady-state simulation optimization is that whenever a solution θ is sampled for the k^{th} time, it suffices to simulate the sample path of X_θ from time t_{k-1} to t_k starting from the state $X_\theta(t_{k-1})$, where the sample path of X_θ was stopped when the solution θ was sampled for $(k-1)^{th}$ time. This requires storing the state of the system and all other

information required for continuing the simulation of a sample path each time when the simulation of that sample path is stopped for each feasible solution. This way of simulating sample paths yields highly accurate objective function estimates for a given simulation budget, but usually comes at the cost of higher storage requirements and longer times necessary to initialize a simulation run. Addressing this tradeoff is beyond the scope of the present work. For additional discussion on this issue, the interested reader is referred to Hong and Nelson [50].

Next we present another way to satisfy Assumption 3.1 in the case of steady-state simulation. In particular, when an objective function estimate is needed at a solution θ for the k^{th} time, we obtain N_k independent realizations of the stochastic process X_θ from time 0 to t_k, i.e., $\{X_\theta^i\}_{i=1}^{N_k}$. Let $\{t_k\}_{k=1}^{\infty}$ be a sequence of positive numbers such that $t_k \to \infty$ as $k \to \infty$ and $\hat{f}_k(\theta) = \sum_{i=1}^{N_k} \int_0^{t_k} h_\theta(X_\theta^i(u))\,\mathrm{d}u/(N_k t_k)$. Then Assumption 3.1 can be easily satisfied (see, e.g., Section 3.1 in Homem-de-Mello [49] and the proof of Theorem 3.2 in Andradóttir [12]).

We say that $a_n = o(b_n)$ if $a_n/b_n \to 0$ as $n \to \infty$. Suppose that all random elements in the R-BEESE algorithm (ones needed for sampling solutions and simulating their performance) are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We now present our convergence analysis for the R-BEESE method.

Theorem 3.2. *Suppose that Assumption 3.1 holds, $|\Theta| < \infty$, and $M_n = o(n)$. Also assume that $G(\{\theta\}) \geq \epsilon > 0$ for all $\theta \in \Theta$. Then the R-BEESE method converges almost surely to the set of optimal solutions $\Theta^* = \{\theta \in \Theta : f(\theta) \geq f(\theta') \text{ for all } \theta' \in \Theta\}$; i.e., for almost every $\omega \in \Omega$, there exists an iteration number $N(\omega)$ such that $\theta_n^*(\omega) \in \Theta^*$ for all $n \geq N(\omega)$.*

Proof: Let $\Omega_1 \subset \Omega$ be such that $\liminf_{n\to\infty} C_n(\theta)/n \geq \epsilon p(1-\alpha)$ and $\lim_{k\to\infty} \hat{f}_k(\theta) = f(\theta)$ for all $\theta \in \Theta$. Since $|\Theta| < \infty$ and $G(\{\theta\}) \geq \epsilon$ for all $\theta \in \Theta$, Assumption 3.1 implies that $\mathbb{P}(\Omega_1) = 1$. Fix $\omega \in \Omega_1$ and let $\gamma = \max_{\theta\in\Theta} f(\theta) - \max_{\theta\in\Theta\setminus\Theta^*} f(\theta) > 0$. Then there exists an iteration number $N(\omega)$ such that for all $n \geq N(\omega)$ and $\theta \in \Theta$, we have that $C_n(\theta, \omega) \geq M_n$ and $|f_n(\theta, \omega) - f(\theta)| < \gamma/2$. Hence, $\theta_n^*(\omega) \in \Theta^*$ for all $n \geq N(\omega)$. ■

Remark 3.2. Similar extensions to Remark 3.1 are also possible for the R-BEESE method

except that we now assume that $p_n \geq p > 0$ and $G_n(\{\theta\}) \geq \epsilon > 0$ for all $\theta \in \Theta$. Moreover, the parameters α and m can be chosen adaptively in each iteration based on the information gathered by the algorithm so far as long as α is uniformly bounded away from 1 and $m \geq 1$ with positive probability whenever a global sampling distribution is used.

3.4 The Adaptive BEES and BEESE methods

3.4.1 Deterministic Optimization Using the A-BEES Method

In this section, we present the A-BEES method for deterministic optimization. Whereas R-BEES samples alternatives randomly either from local or global distributions, the A-BEES method adaptively alternates between sampling from local or global distributions with the goal of using the "appropriate" (local or global) distribution at each stage of the search. As before, the global distribution G aims at exploring the entire feasible region, while the family of local distributions $\mathcal{L}$ aims at searching promising subregions (exploitation).

More specifically, after sampling k points since the last review (decision about the nature of the search), a decision is made about whether the next k sampled points will be selected using local or global sampling distributions. Let v^* be the function value of the best solution θ_n found so far and v_l^* be the function value at the best point found the last time local search was performed. Let Δ be the improvement in the function value between the current and preceding reviews and D be the distance between the points where the corresponding function values were achieved. The pseudo-code for how the method alternates between sampling distributions is given in Algorithm 3.2. If the flag LocalSearch is true, then the method performs local search; otherwise, it does global search. Observe that Algorithm 3.2 requires two thresholds, namely the distance threshold d and the improvement threshold δ.

We now motivate the sampling distribution update procedure. The method switches from local to global search only if the improvement in the objective function value between successive reviews is small (less than δ). Usually this means that the local search has identified a locally optimal solution (or a solution with a near-local-optimal objective function value) and hence there is little merit in continuing searching locally. On the other hand, if local search is making good progress, then the method will continue searching locally.

Algorithm 3.2 (Sampling Distribution Update Procedure)

```
1: if LocalSearch=true then
2:   if Δ ≤ δ then
3:     LocalSearch ← false
4:     v_l^* ← v^*
5:   end if
6: else
7:   if Δ ≤ δ then
8:     if v^* − v_l^* ≥ δ then
9:       LocalSearch ← true
10:    end if
11:  else
12:    if D ≤ d then
13:      LocalSearch ← true
14:    end if
15:  end if
16: end if
```

The A-BEES method can switch from global to local search in two ways. The first way occurs when the improvement Δ is small but substantial improvement in the objective function value has been achieved since the last switch from local to global search. This means that the method has identified a promising region and global search is not yielding substantial progress. Hence, local search can be more beneficial (due to problem structure) at this stage of the search. The second way occurs when the improvement between successive reviews is large but the distance D is small. This makes sense because the improvement has been local in nature and hence switching to local search may be beneficial.

Note that a practitioner has a lot of flexibility in defining the distance D without affecting the convergence guarantee of the A-BEES method. The distance D from θ_1 to θ_2 can be calculated, for example, based on some metric (e.g., the Euclidean distance if $\Theta \subset \mathbb{R}^d$). Observe that if a certain dimension of the search space Θ is deemed more important than other dimensions (for example, when objective function values change faster in this dimension than in the others), then this dimension can be given larger weight in calculating the distance. Also, the following notion of distance can be useful, especially if the feasible region is combinatorial. For each $\theta \in \Theta$, let $N(\theta) \in \Theta$ be a set of local neighbors of θ. Let F be the neighborhood graph induced by $\{N(\theta)\}_{\theta \in \Theta}$. Then D can be the shortest distance from θ_1 to θ_2 with each arc in the graph F having weight 1. This distance can be evaluated

using either Dijkstra's algorithm or the Bellman-Ford algorithm. Also, observe that it is not necessary to evaluate D, because it suffices to decide if $D \leq d$. This is usually easier to accomplish (for instance, Dijkstra's algorithm can be terminated with the answer that $D > d$ if the next permanent node has distance $d+1$ from θ_1 and θ_2 is not one of the permanent nodes). The pseudo-code for the A-BEES algorithm is given in Algorithm 3.3.

Algorithm 3.3 (A-BEES Algorithm)

counter $\leftarrow 0$, $n \leftarrow 0$
LocalSearch $\leftarrow$ false
Sample a solution θ from the global distribution G
Evaluate the objective function at θ
Let $v^*, v_l^* \leftarrow f(\theta)$ and $\theta_n \leftarrow \theta$
while Stopping criterion is not satisfied **do**
  if LocalSearch=true **then**
    Sample a solution θ from a local distribution in $\mathcal{L}$
  else
    Sample a solution θ from the global distribution G independent of everything else
  end if
  Evaluate the objective function at θ (if needed)
  counter $\leftarrow$ counter+1, $n \leftarrow n+1$
  Update θ_n and v^* (if needed)
  if counter=k **then**
    Update Δ and D
    Update search nature (use Sampling Distribution Update Procedure)
    counter $\leftarrow 0$
  end if
end while
Present $\theta_n^* = \theta_n$ as the estimate of the optimal solution

Suppose that all random elements in the A-BEES method (ones needed for sampling solutions) are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Our convergence result for the A-BEES method for deterministic optimization is given in Theorem 3.3.

Theorem 3.3. *Under the conditions in Theorem 3.1, we have* $\lim_{n\to\infty} f(\theta_n^*) = f^*$ *with probability one.*

Proof: (i) First we show that the A-BEES method uses the global sampling distribution G i.o. for every $\omega \in \Omega$. Contrary to this, suppose that there exists an $\omega \in \Omega$ such that the global sampling distribution is used a finite number of times. Then there exists an iteration number $N_0(\omega)$ such that local sampling distributions are used from this iteration

on. Let $\bar{f}(\omega)$ be the objective function value at the best point found prior to iteration $N_0(\omega)$. Observe that the maximum number of successive reviews for which the local search will be continued after iteration $N_0(\omega)$ is $\lfloor (f^* - \bar{f}(\omega))/\delta \rfloor + 1 < \infty$, where $\lfloor \cdot \rfloor$ is the floor function. This provides a contradiction. The remainder of the proof is identical to the proof of part (i) of Theorem 3.1 except that Ω_k^n is defined to be the subset of Ω such that the A-BEES method samples a solution in the set A_k when the global sampling distribution G is used for the n^{th} time for all $k \in \mathbb{N} \setminus \{0\}$ and $n \in \mathbb{N}$, so that $\mathbb{P}(\Omega_k^n) = G(A_k) > 0$. This concludes the proof of (i).

(ii) Let $\tilde{\Omega}_1 \subset \Omega$ be such that global sampling is performed a finite number of times. Fix $\omega \in \tilde{\Omega}_1$. Since the only reason for not switching back to global search is that the objective function value improvement between successive reviews is at least δ, we conclude that $f(\theta_n^*)$ goes to infinity for this ω. It remains to show that $f(\theta_n^*)$ diverges to infinity for almost every $\omega \in \tilde{\Omega}_2 = \Omega \setminus \tilde{\Omega}_1$ (the set of ω's such that the global sampling distribution is used i.o.).

Fix $l \in \mathbb{N} \setminus \{0\}$ and $\omega \in \tilde{\Omega}_2 \bigcap \tilde{\Omega}'$, where $\tilde{\Omega}'$ is the set $\tilde{\Omega}$ of part (i) of the proof of Theorem 3.1 with the exceptions that A_k is substituted by B_k and Ω_k^n is defined as in part (i) of this proof. Then there exists an iteration number $N_l(\omega)$ such that some solution in the set B_l is sampled at iteration $N_l(\omega)$. As l is arbitrary, we get that $f(\theta_n^*) \to f^*$ as $n \to \infty$ for this ω.

Observe that $\mathbb{P}(\tilde{\Omega}') = 1$ (the proof of this is similar to the proof that $\mathbb{P}(\tilde{\Omega}) = 1$ in part (i) of Theorem 3.1) which implies that $\mathbb{P}(\tilde{\Omega}_2 \bigcap \tilde{\Omega}') = \mathbb{P}(\tilde{\Omega}_2)$, and the proof is complete. ■

Remark 3.3. If $f^* < \infty$, then with probability one after some iteration number (depending on the particular sample path of the A-BEES method) only global search will be conducted. This is desirable because the promising subregions identified so far have already been sampled thoroughly using local search and consequently substantial improvement can only be accomplished if a new promising subregion is identified using global search.

Remark 3.4. Similarly to Remark 3.1 the A-BEES method can also be made highly adaptive.

3.4.2 Stochastic Optimization Using the A-BEESE Method

In this section, we describe how the A-BEES method of Section 3.4.1 is adapted to handle stochastic optimization problems and discuss the convergence of the resulting A-BEESE method. For each $\theta \in \Theta$, let $f_n(\theta)$ be defined as in Section 3.3.2 and suppose that two successive reviews happen in iterations n_1 and n_2, where $n_1 < n_2$. Let θ_n be the solution with the highest estimated objective function value in iteration n (with ties broken arbitrarily). Then D is the distance from θ_{n_1} to θ_{n_2} and $\Delta = f_{n_2}(\theta_{n_2}) - f_{n_2}(\theta_{n_1})$. Similarly, $v^* = f_{n_2}(\theta_{n_2})$ and $v_l^* = f_{n_2}(\theta_l)$, where l is the last iteration number in which local search was performed.

Now we present the modifications of the A-BEES method that are designed to incorporate features that aid estimation. First, in iteration n, steps 7 through 11 of Algorithm 3.3 are executed with probability $1-\alpha$ and with probability $0 \leq \alpha < 1$ the point θ_n is sampled. Secondly, we switch to local search if global search has been conducted for g consecutive reviews (this is a modification to Algorithm 3.2). Observe that as the search progresses, this modification forces the method to perform estimation (because local search toward the end of the search samples points that have already been seen and have high estimated objective function values under our structural assumption). Finally, in the case of transient simulation, we conduct m simulation replications at a solution whenever it is sampled (see Section 3.3.2 for two ways of performing simulation runs in steady-state settings).

The next modification to the A-BEES method can also be viewed as a generalization (and hence, can also be applied in deterministic optimization). We conduct local (global) search for k_l (k_g) iterations before attempting to switch to global (local) search (by invoking Algorithm 3.2). Typically, the parameters k_l and k_g satisfy $k_l \geq k_g$. This modification might not be extremely helpful for deterministic optimization because, as has been mentioned in Section 3.2, local search acts more like an estimation component in the later stages of the search and we do not need estimation when objective function values can be evaluated without noise. This is also not necessary in the early stages of the search because the A-BEES method stays local if local search has been making good progress.

The final modification to the A-BEES method is that as in the R-BEESE method,

the optimal solution θ_n^* is estimated using the estimator proposed in Andradóttir [14], see Section 3.3.2 for more details. Note that in case of deterministic optimization, A-BEESE with $\alpha = 0$, $m = 1$, $M_n = 1$, $g = +\infty$, and $k_l = k_g$ reduces to the A-BEES method.

Suppose that all random elements in the A-BEESE method (needed for sampling solutions and evaluating their performance) are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We will need the following result to prove the convergence of the A-BEESE method.

Lemma 3.1. *Suppose that Assumption 3.1 holds and that $|\Theta| < \infty$. Then the A-BEESE algorithm uses the global sampling distribution G infinitely often with probability one.*

Proof: Let $\Omega_1 \subset \Omega$ be such that Assumption 3.1 holds for all $\theta \in \Theta$. Because Θ is finite, we have that $\mathbb{P}(\Omega_1) = 1$. It suffices to show that the global distribution G is used i.o. for every $\omega \in \Omega_1$. Suppose that there exists an $\omega \in \Omega_1$ for which G is used only a finite number of times. Let $\Theta(\omega) \subset \Theta$ be the set of alternatives that are sampled infinitely often. Since Θ is finite, there exists an iteration number $N_1(\omega)$ such that for all $n \geq N_1(\omega)$, local search is performed in iteration n, the algorithm samples points in the set $\Theta(\omega)$ in iteration n, and we have $|f_n(\theta) - f(\theta)| < \delta/4$ for all $\theta \in \Theta(\omega)$. Note that for any iterations $n_1, n_2 \geq N_1(\omega)$ and $\theta \in \Theta$, we have that $|f_{n_1}(\theta) - f_{n_2}(\theta)| < \delta/2$. Now consider two successive search reviews at iterations n_1 and n_2 ($n_1 < n_2$) that occur after iteration $N_1(\omega)$. We have that

$$\Delta = f_{n_2}(\theta_{n_2}) - f_{n_2}(\theta_{n_1}) \leq |f_{n_2}(\theta_{n_2}) - f_{n_1}(\theta_{n_2})| + [f_{n_1}(\theta_{n_2}) - f_{n_1}(\theta_{n_1})] + |f_{n_1}(\theta_{n_1}) - f_{n_2}(\theta_{n_1})| < \delta,$$

where we have used the definition of θ_{n_1}. Since $\Delta < \delta$, the algorithm switches from local to global search. This provides a contradiction and hence the proof is complete. ■

We are now ready to state and prove our convergence result for the A-BEESE method.

Theorem 3.4. *Under the same conditions as in Theorem 3.2, the A-BEESE method converges almost surely to the set of optimal solutions Θ^*.*

Proof: Observe that the random elements in the A-BEESE method are of four types, namely, the ones needed for estimating the objective function value at each solution (defined on Ω_s), the ones needed for sampling solutions using local sampling distributions in

$\mathcal{L}$ (defined on Ω_l), the ones needed for sampling solutions using the global sampling distribution G (defined on Ω_g), and the ones needed to decide whether to sample the point with the highest estimated objective function value or use the current sampling distribution (defined on Ω_a). Hence, without loss of generality, we assume that $\Omega = \Omega_s \times \Omega_l \times \Omega_g \times \Omega_a$. Also, without loss of generality, we assume that $\Omega_g = \prod_{i=0}^{\infty} \Omega_g^i$, where Ω_g^i is a sample space on which random elements for sampling from the global distribution G for the i^{th} time are defined.

Let $\tilde{\Omega}_s \subset \Omega_s$ be such that Assumption 3.1 holds for each $\theta \in \Theta$. Because $|\Theta| < \infty$, we have that $\mathbb{P}(\tilde{\Omega}_s) = 1$. For each $\theta \in \Theta$, $l \in \mathbb{N}\backslash\{0\}$, and $\omega_g \in \Omega_g$, let $H_l(\theta, \omega_g)$ be the number of times a solution θ has been sampled by the time the global distribution G has been used for the l^{th} time (we also let $H_l(\theta, \omega) = H_l(\theta, \omega_g)$, where $\omega = (\omega_s, \omega_l, \omega_g, \omega_a) \in \Omega$). Let $\tilde{\Omega}_g \subset \Omega_g$ be such that $\liminf_{l\to\infty} H_l(\theta, \omega_g)/l \geq \epsilon$ for all $\theta \in \Theta$. Since $G(\{\theta\}) > \epsilon$ for all $\theta \in \Theta$ and Θ is finite, the Strong Law of Large Numbers implies that $\mathbb{P}(\tilde{\Omega}_g) = 1$. Let $\tilde{\Omega}_a \subset \Omega_a$ be such that the long-run average fraction of time the point with the highest estimated objective function value is sampled equals α. By the Strong Law of Large Numbers we have that $\mathbb{P}(\tilde{\Omega}_a) = 1$.

Fix $\omega \in \tilde{\Omega}_s \times \Omega_l \times \tilde{\Omega}_g \times \tilde{\Omega}_a$ and let $\gamma = \max_{\theta\in\Theta} f(\theta) - \max_{\theta\in\Theta\backslash\Theta^*} f(\theta) > 0$. Also define $\xi = \min(\gamma/2, \delta/4)$. By Lemma 3.1, the finiteness of Θ, and the choice of ω, it follows that there exists an iteration number $N_2(\omega)$ such that for all $\theta \in \Theta$ and $n \geq N_2(\omega)$, we have that $|f_n(\theta, \omega) - f(\theta)| < \xi$ (note that $\tilde{\Omega}_s = \Omega_1$ in the proof of Lemma 3.1). Recall that $C_n(\theta)$ denotes the number of times a solution θ has been simulated by iteration n. Then it suffices to show that there exists $N(\omega)$ such that $C_n(\theta, \omega) \geq M_n$ for all $\theta \in \Theta$ and $n \geq N(\omega)$.

Let $l_n(\omega)$ be the number of times the global distribution G is used by iteration n. Since $f_n(\theta)$ is within $\delta/4$ of $f(\theta)$ for all $\theta \in \Theta$ and $n \geq N_2(\omega)$, the improvement in the estimated objective function values between iterations n_1 and n_2 will be less than δ for all $n_1, n_2 \geq N_2(\omega)$. This implies that after some iteration $N_3(\omega)$, the A-BEESE method will perform $k_g * g$ iterations of global search followed by k_l iterations of local search, and then the cycle will repeat. Taking into account the sampling of θ_n with probability α, this

implies that

$$\lim_{n\to\infty} \frac{l_n(\omega)}{n} = (1-\alpha)\frac{gk_g}{k_l + gk_g} > 0.$$

Then, for all $\theta \in \Theta$,

$$\begin{aligned}
\liminf_{n\to\infty} \frac{C_n(\theta,\omega)}{n} &= \liminf_{n\to\infty} \frac{C_n(\theta,\omega)}{l_n(\omega)} \times \lim_{n\to\infty} \frac{l_n(\omega)}{n} \\
&\geq \liminf_{n\to\infty} \frac{H_{l_n(\omega)}(\theta,\omega)}{l_n(\omega)} \times (1-\alpha)\frac{gk_g}{k_l + gk_g} \\
&\geq \epsilon(1-\alpha)\frac{gk_g}{k_l + gk_g} > 0.
\end{aligned}$$

Because $M_n = o(n)$, this implies that there exists an iteration number $N(\omega)$ so that $C_n(\theta,\omega) \geq M_n$ for every $\theta \in \Theta$ and all $n \geq N(\omega)$, and hence the proof is complete. ■

Remark 3.5. Remark 3.2 with obvious modifications is also valid for A-BEESE, and hence A-BEESE can be made even more adaptive without affecting its convergence guarantee.

3.5 Numerical examples

In this section, we use numerical results to analyze our algorithms and to support the discussion of Section 3.2. More specifically, in Section 3.5.1, we describe the test problems used in our numerical experiments. In Section 3.5.2, we provide numerical results that validate the discussion given in Section 3.2. We also compare the empirical performance of the proposed methods to that of other algorithms in Section 3.5.3 and evaluate the performance of the optimization methods under two different estimators of the optimal solution in Section 3.5.4.

3.5.1 Test Problems

In this section, we describe our three test problems. The first problem is referred to as the *unimodal* problem. It is given by

$$\Theta = \{(i,j) \in \mathbb{N}^2 : 0 \leq i,j \leq 199\},$$

$$f(\theta_1,\theta_2) = \max\{0, -(\theta_1 - 30)^2 - (\theta_2 - 30)^2 + 400\},$$

and $h_\theta(X_\theta) = f(\theta) + X_\theta$, where X_θ is a $N(0, \sigma^2)$ random variable for each $\theta \in \Theta$, where $N(\mu, \sigma^2)$ denotes a normal random variable with mean μ and variance σ^2. This problem has a relatively large feasible region (with $40,000$ solutions) and only a very small proportion of solutions with high objective function values. It models optimization problems where a small fraction of solutions have substantially better performance than other solutions.

The second test problem is the *two hills* problem. It is given by

$$\Theta = \{\theta = (x_1, x_2) \in \mathbb{N}^2 : 0 \leq x_1, x_2 \leq 49\},$$

$$f(\theta) = \max\{f_1(\theta), f_2(\theta), 0\},$$

and $h_\theta(X_\theta) = f(\theta) + X_\theta$, where

$$f_1(\theta) = -(0.4x_1 - 5)^2 - 2(0.4x_2 - 17.2)^2 + 7,$$

$$f_2(\theta) = -(0.4x_1 - 12)^2 - (0.4x_2 - 4)^2 + 4,$$

and X_θ is a $N(0, \sigma^2)$ random variable for each $\theta \in \Theta$. This problem is of interest because its objective function has two hills of different heights (4 and 6.96), located relatively far apart (the hill of height 4 is centered at $(30, 10)$ and the hill of height 6.96 is centered at $(12, 43)$ and $(13, 43)$), and separated by a flat valley (of height 0). This problem is useful for testing how the proposed methods behave on problems with multiple locally optimal solutions. This problem was also used in the numerical studies of Prudius and Andradóttir [73, 74].

The last test problem is the *three-stage buffer allocation* problem. This is a three-stage flow line with an infinite supply of jobs in front of station 1 and a finite buffer capacity in front of station 2 and 3. Production blocking is assumed; i.e., if the buffer in front of station k is full, then the completed unit at station $k-1$ can not be released, and hence, station $k-1$ gets blocked. The goal is to identify an allocation of buffers and service rates such that the long-run average throughput is maximized, subject to limited total buffer capacity and service rates. Service times at each station are independent and exponentially distributed. Let x_k be the service rate at station $k \in \{1, 2, 3\}$, and x_4 and x_5 be the buffer capacities in

front of stations 2 and 3, respectively. The feasible region is given by

$$\Theta = \{(x_1, \ldots, x_5) : x_1 + x_2 + x_3 \leq 20; x_4 + x_5 = 20; 1 \leq x_k \leq 20, x_k \in \mathbb{N}^+, k = 1, \ldots, 5\}.$$

The cardinality of the feasible region is $21,660$. The balance equations for the underlying Markov chain can be obtained from Buzacott and Shantikumar [26] and these equations can be solved numerically. Thus, the expected throughput can be calculated explicitly for each feasible configuration. The optimal throughput is 5.776, which is attained at two feasible solutions. In the deterministic three-stage buffer allocation problem, the objective function values are evaluated without noise (i.e., by solving the linear system of equations). In the stochastic version of this problem, the throughput is estimated after the first $2,000$ units have been released, and it is averaged over the next 50 units produced (i.e., we consider a transient approximation of the problem). This simulation optimization problem was used in the numerical experiments of Pichitlamken and Nelson [71] and hence the results for this problem provide a limited empirical comparison of the approaches (a more extensive comparison of these approaches is beyond the scope of the present work).

3.5.2 BEES(E) Framework

The numerical experiments in this section use the R-BEES and R-BEESE methods to support the ideas discussed in Section 3.2 pertaining to exploration, exploitation, and estimation. We first address the issue of exploration and exploitation in deterministic optimization. Consider a deterministic version of the unimodal problem, so that $\sigma^2 = 0$. We let the global sampling distribution G be a uniform distribution on Θ and the family of local sampling distributions $\mathcal{L}$ consist of uniform distributions on each set in $\{N_0(\theta) : \theta \in \Theta\}$, where

$$N_0(\theta) = \{(x_1, x_2) \in \Theta \setminus \{\theta\} : |x_i - \theta_i| \leq 1 \text{ for } i = 1, 2\}$$

for all $\theta \in \Theta$. If local search is performed in iteration n, the method employs a uniform distribution on $N_0(\theta_n)$ (recall that θ_n is a solution with the highest objective function value found so far), so that the search is local in the best subregion found so far. We use the R-BEES method with parameter $p = 1$ (so that the method performs only exploration), $p = 0.3$ (so that the method does some exploitation), and $p = 0$ (so that the method

does only exploitation). Figure 3.2 shows the average performance of 100 independent replications of the three approaches. From Figure 3.2 it is obvious that initially the R-BEES method with parameter $p = 1$ has better empirical performance than R-BEES with $p = 0.3$, but as the simulation effort increases the situation reverses. Moreover, R-BEES with $p = 0$ is by far the worst. This supports the ideas discussed in Section 3.2 that exploration is most beneficial at the beginning of the search, with exploitation becoming more effective as the search progresses. Moreover, given the difficulties associated with determining when the focus should be shifted from exploration to exploitation (see Section 3.2), this figure also suggests that it is desirable to maintain appropriate balance between the two throughout the search.

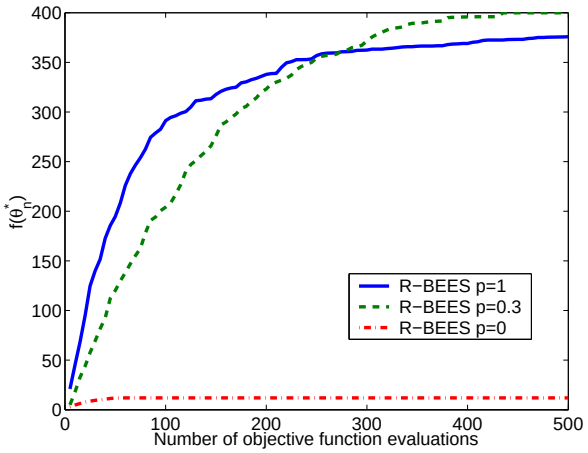


Figure 3.2: Performance of the R-BEES method on the unimodal problem with $\sigma^2 = 0$

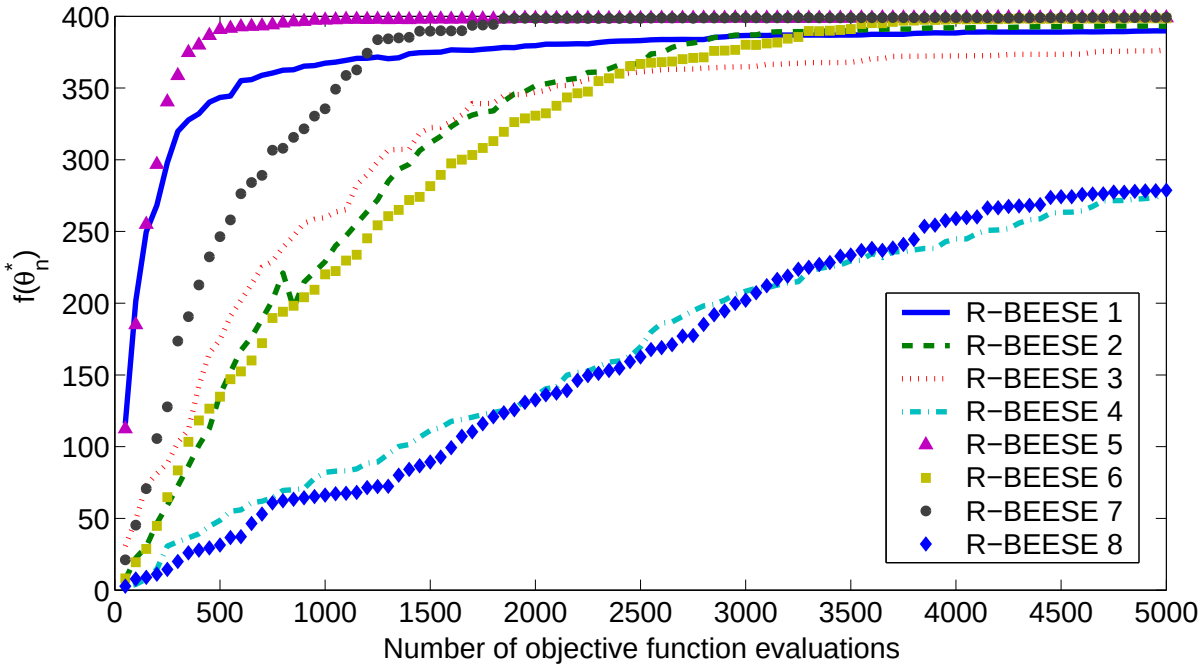
The next experiment is performed to explain the role of estimation in simulation optimization. For this experiment we use the R-BEESE method with $M_n = \sqrt{n}$ for all n, the global distribution G being the uniform distribution on Θ, and the family of local distributions being defined and used as in the preceding example. We are interested in evaluating the performance of the R-BEESE method as a function of the three parameters that aid estimation, namely, α, p, and m, and let R-BEESE $(\tilde{\alpha}, \tilde{p}, \tilde{m})$ refer to the R-BEESE method with the specifications above and with the values of α, p, and m being $\tilde{\alpha}$, $\tilde{p}$, and $\tilde{m}$, respectively. We consider high (low) noise problems where the standard deviations of single objective function observations are of the order of (a few orders of magnitude smaller than)

the range of the objective function values. Figure 3.3 shows the average performance over 100 independent replications of the R-BEESE method with different parameter values on the unimodal problem with $\sigma^2 = 1,000$ (low noise) and $\sigma^2 = 160,000$ (high noise). The abbreviation for each method is given in Table 3.1. Observe that the experiment is a full factorial design with two levels for the parameters α, p, and m.

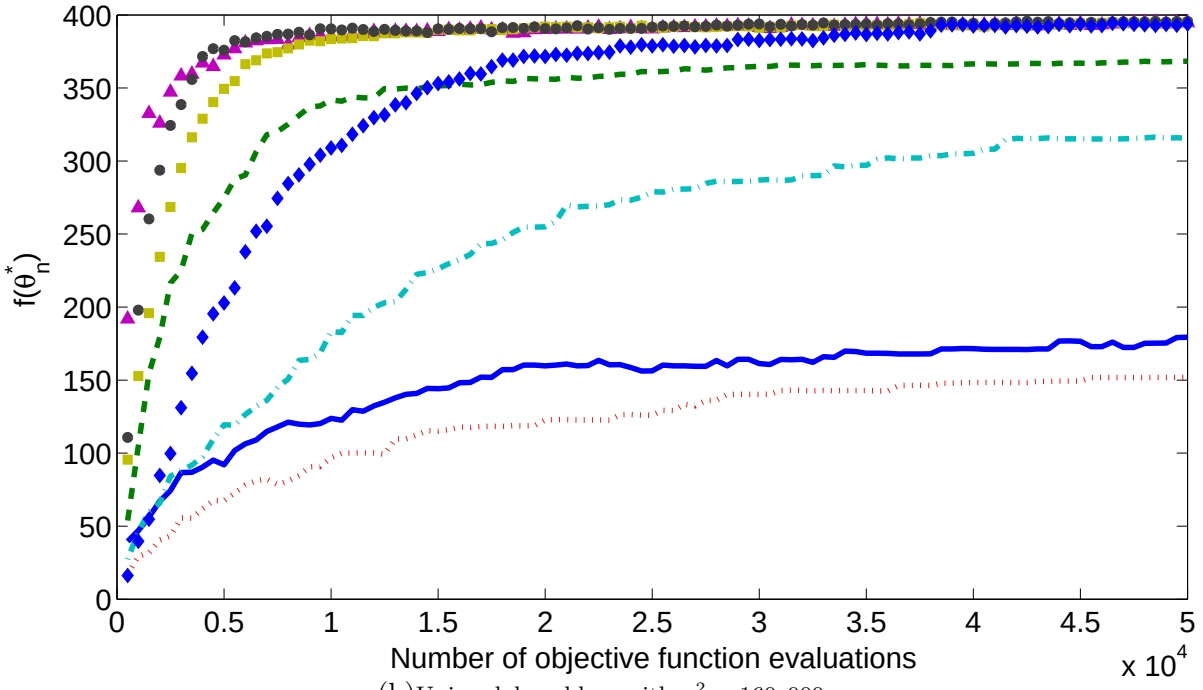
Table 3.1: Abbreviation for the R-BEESE methods in Figure 3.3

R-BEESE 1	R-BEESE $(0, 0.5, 1)$
R-BEESE 2	R-BEESE $(0, 0.5, 10)$
R-BEESE 3	R-BEESE $(0, 0.1, 1)$
R-BEESE 4	R-BEESE $(0, 0.1, 10)$
R-BEESE 5	R-BEESE $(0.3, 0.5, 1)$
R-BEESE 6	R-BEESE $(0.3, 0.5, 10)$
R-BEESE 7	R-BEESE $(0.3, 0.1, 1)$
R-BEESE 8	R-BEESE $(0.3, 0.1, 10)$

From Figure 3.3, it is easy to see that the probability α of resampling the solution with the highest estimated objective function value has the greatest impact on the performance of the R-BEESE method. Observe that for the high noise problem, the R-BEESE methods 5 through 7 (with $\alpha = 0.3$) perform better the R-BEESE methods 1 through 4 (with $\alpha = 0$), and R-BEESE 8 eventually becomes better than the R-BEESE methods 1 through 4. This supports the idea that for high noise settings, the parameter α is crucial for estimating the optimal solution toward the end of the search. This also shows that it is more effective to do estimation through resampling the best point and conducting local search around the best point, rather than by increasing simulation effort at every point. This makes sense because we are targeting estimation effort to where it is needed, rather than by doing it indiscriminately. For the low noise setting, R-BEESE 5 performs better than the rest of the approaches, and eventually R-BEESE 7 becomes the second best (both methods have $\alpha = 0.3$ and $m = 1$). Consequently R-BEESE 5 has good performance for both noise settings. Observe also that too much estimation is not good as can be seen from part (a) of Figure 3.3, where R-BEESE 8 performs most estimation and is nearly the worst method. From these numerical studies, we conclude that in order to attain good empirical



(a) Unimodal problem with $\sigma^2 = 1,000$



(b)Unimodal problem with $\sigma^2 = 160,000$

Figure 3.3: Performance of the R-BEESE method on the unimodal problem with $\sigma^2 = 1,000$ and $\sigma^2 = 160,000$

performance, it is important to perform estimation at a level suitable to the problem at hand and that α is the most important parameter to control (with m being the second most

important parameter if $\alpha = 0$). The idea of controlling α for estimation purposes is novel because most random search methods available in the literature do this by adjusting the parameter m.

The same numerical experiment was conducted on the two hills problem with $\sigma^2 = 1$ (low noise) and $\sigma^2 = 50$ (high noise). The results were similar to Figure 3.3 and are omitted to conserve space. This suggests that the two hills and smaller feasible region of the two hills problem balance out with the single hill and large feasible space of the unimodal problem.

3.5.3 Algorithm Comparison

In this section we compare the performance of the R-BEES(E) and A-BEES(E) methods to that of the SA algorithm with constant temperature of Alrefaei and Andradóttir [3]. The numerical performance of our search methods is also compared to that of the NP method of Pichitlamken and Nelson [71] on the three stage buffer allocation problem.

In what follows, Global SA refers to Algorithm 2 in Alrefaei and Andradóttir [3] with neighborhood structure given by $\Theta \setminus \{\theta\}$, while Local SA refers to the same algorithm with neighborhood structure $N_0(\theta)$ for the two hills and unimodal problems and $N_1(\theta)$ for the three stage buffer allocation problem, where θ is the current solution and $N_1(\theta)$ is the set of feasible points that can be obtained by shifting a single buffer slot between buffers, increasing or decreasing a service rate by 1 at a single workstation, or shifting a single unit of service rate between two workstations. In the R-BEES(E) and A-BEES(E) methods, the global sampling distribution is a uniform distribution on Θ, while the local sampling distribution is a uniform distribution on $N_0(\theta_n)$ ($N_1(\theta_n)$) for the two hills and unimodal problems (three stage buffer allocation problem), where θ_n is a solution with the highest estimated objective function value. The distance D in our experiments is Euclidean.

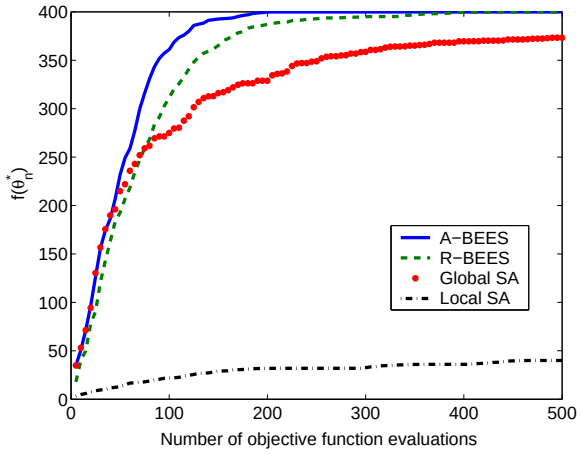
An effort was made to select good parameter values for each algorithm (the parameter values for each method were optimized over a set of substantially different values as explained below). In particular, the values of p for R-BEES(E) and $k = k_l = k_g, \delta, d, g$ for A-BEES(E) on each problem were optimized for one particular level of noise (depending on the problem) and used for all noise levels of this problem, while α, p, and M_n were set to

smaller values for low noise problems and larger values for high noise problems. Similarly, the value of the temperature T for Global and Local SA was optimized for each problem and the parameter L_k (number of objective function observations collected at the current and candidate solutions in iteration k) is picked similarly to m. The resulting parameter values are given in Table 3.2. The performance of the algorithms is averaged over 100 independent replications for the unimodal and two hills problems and over 50 independent replications for the three stage buffer allocation problem (simulation runs for the stochastic version of the three stage buffer allocation problem are computationally expensive).

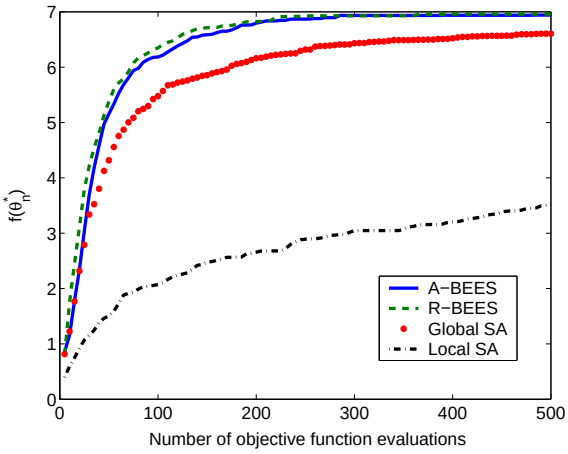
Table 3.2: Parameter values for each algorithm

	Problem	Two Hills			Unimodal			Buffer Allocation	
	σ^2	0	1	50	0	1,000	160,000	Det.	Stoch.
Local SA	L_k	1	1	10	1	1	10	1	3
	T	0.1	5	5	1	1	1	0.1	0.1
Global SA	L_k	1	1	10	1	1	10	1	3
	T	1	1	5	1	1	10	1	0.5
R-BEESE	p	0.8	0.8	0.8	0.7	0.7	0.7	0.4	0.4
	α		0.3	0.3		0.3	0.3		0.3
	m		1	10		1	10		3
	M_n		$\sqrt{n}$	$\sqrt{n}$		$\sqrt{n}$	$\sqrt{n}$		$\sqrt{n}$
A-BEESE	k, k_l, k_g	10	10	10	20	20	20	10	10
	δ	0.05	0.05	0.05	20	20	20	0.01	0.01
	d	5	5	5	5	5	5	5	5
	g		5	5		5	5		2
	α		0.3	0.3		0.3	0.3		0.3
	m		1	10		1	10		3
	M_n		$\sqrt{n}$	$\sqrt{n}$		$\sqrt{n}$	$\sqrt{n}$		$\sqrt{n}$

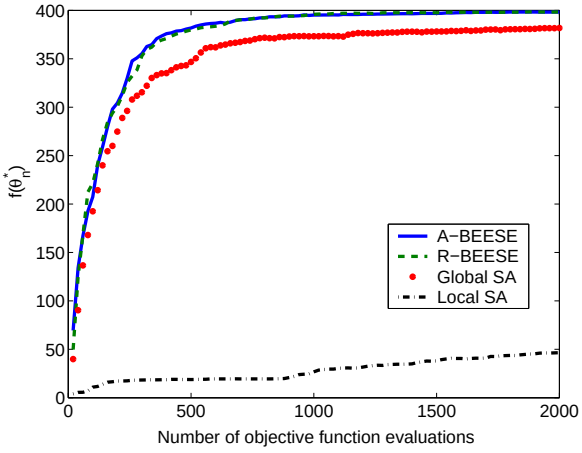
Parts (a) through (c) of Figure 3.4 show the performance of the optimization methods on the unimodal problem. It is obvious that the A-BEES method is considerably better than the R-BEES method when $\sigma^2 = 0$, while the A-BEESE method has similar performance to the R-BEESE method when $\sigma^2 \in \{1,000, 160,000\}$. Global SA performs worse than the A-BEESE and R-BEESE methods when $\sigma^2 \in \{1,000, 160,000\}$, but it is slightly better than the R-BEESE method early in search when $\sigma^2 = 0$ and much worse later in the search. The Local SA algorithm is by far the worst method for this problem. The reason is that the feasible space is large and it might take a while for Local SA to reach a good subregion.



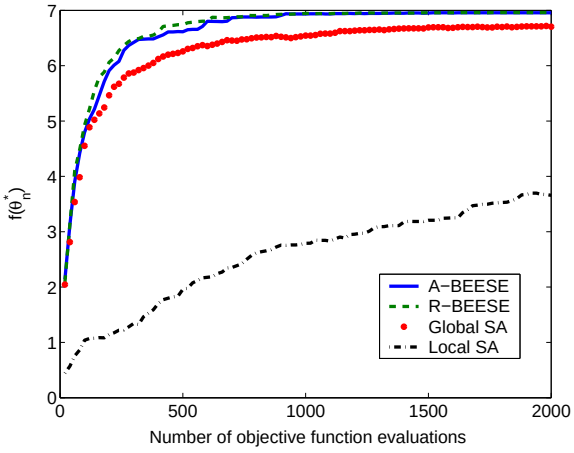
(a) Unimodal problem with $\sigma^2 = 0$



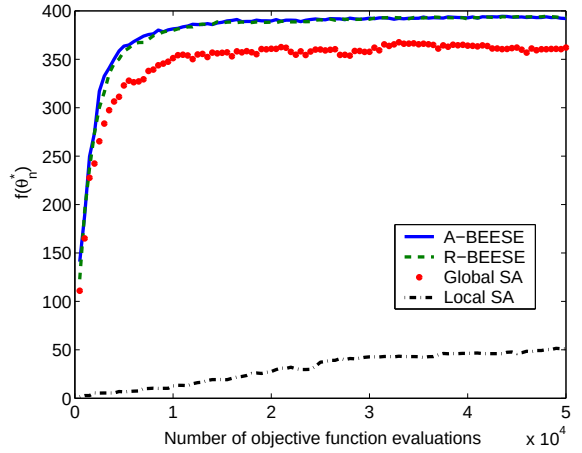
(d) Two hills problem with $\sigma^2 = 0$



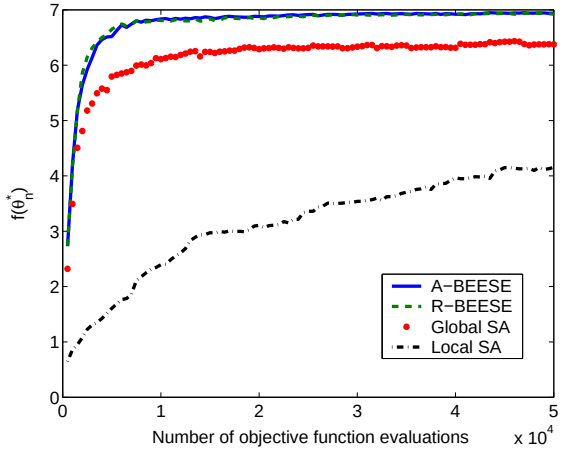
(b) Unimodal problem with $\sigma^2 = 1,000$



(e) Two hills problem with $\sigma^2 = 1$

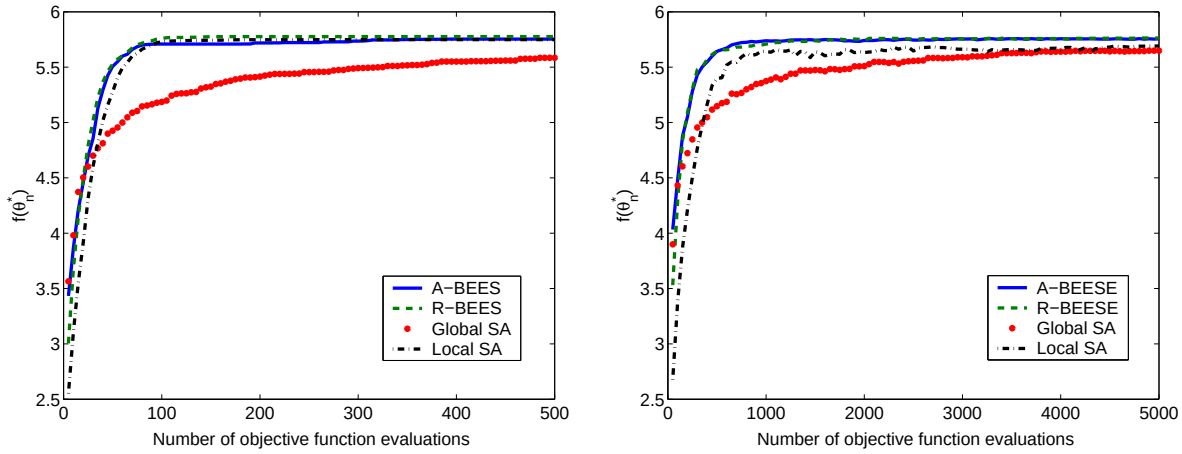


(c) Unimodal problem with $\sigma^2 = 160,000$



(f) Two hills problem with $\sigma^2 = 50$

Figure 3.4: Performance of the A-BEES(E), R-BEES(E), and Local and Global SA methods on the unimodal and two hills problems



(a) Deterministic problem (b) Stochastic problem

Figure 3.5: Performance of the A-BEES(E), R-BEES(E), and Local and Global SA methods on the three stage buffer allocation problem

Parts (d) through (f) of Figure 3.4 show the performance of the various simulation optimization algorithms on the two hills problem. It is clear that the R-BEES(E) method has similar performance to the A-BEES(E) method and these methods are the best on this problem. Global SA is the third best method, while Local SA is by far the worst method. The reason is that this problem has a suboptimal local solution and it is difficult for Local SA to escape from this solution given that it utilizes a local neighborhood structure.

Figure 3.5 shows the performance of the four methods on the three stage buffer allocation problem. On this problem, the A-BEES(E) and R-BEES(E) methods have similar performance, and they outperform both SA algorithms. Global SA is better than Local SA in the early stages of the search but is worse than Local SA in the later stages. From part (b) of Figure 3.5 in this section and Figure 7 in Pichitlamken and Nelson [71], it can be seen that the A-BEESE and R-BEESE methods perform better than the NP methods on this problem (an objective function evaluation is referred to as a "replication" in [71]). But more numerical studies are of course required to adequately compare these approaches.

Figures 3.4 and 3.5 also show that, as expected, the convergence of each method slows down as the noise increases. However, the relative performance of the methods does not depend heavily on the noise level. Moreover, the difference in the empirical performance

of the R-BEESE and A-BEESE methods becomes smaller as σ^2 grows (i.e., noise becomes larger).

From these limited numerical experiments we conclude that the proposed R-BEES(E) and A-BEES(E) methods appear to perform well when compared to algorithms proposed previously in the literature, though more numerical studies are required. The R-BEES(E) and A-BEES(E) methods have similar empirical performance on all the test problems except for the unimodal problem with $\sigma^2 = 0$. In instances where R-BEES(E) and A-BEES(E) methods yield similar performance, the R-BEES(E) method is preferred because the A-BEES(E) method is more complex (in that it has more parameters). We believe that A-BEES(E) is better than R-BEES(E) on problems with large feasible regions, low noise, and small proportions of solutions having high objective function values (the numerical studies above support that, see part (a) of Figure 3.4). The reason for this is that the A-BEES(E) method uses local and global search adaptively and the benefits of selecting the sampling distribution adaptively are more pronounced on problems having the outlined structure (it is easier to identify a proper sampling distribution for problems with low noise and the benefits of doing so are greater for problems with large feasible regions and small proportions of "good" solutions). Again more numerical studies are required to support this conclusion.

3.5.4 Estimation of the Optimal Solution

We conclude this section by investigating the choice of the estimator of the optimal solution. More specifically, we compare two estimators:

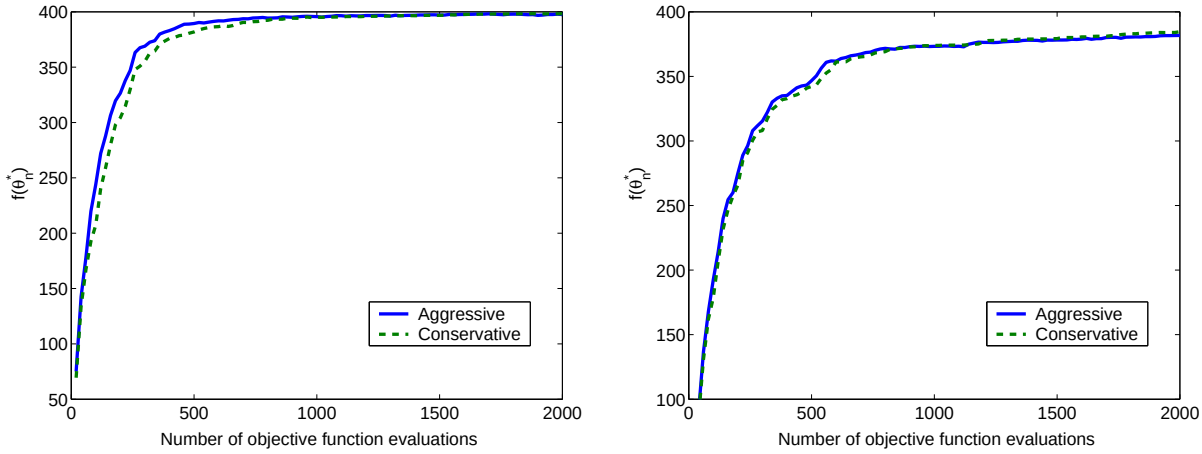
$$\begin{aligned} \theta_n^*(1) &\in \arg\max\{f_n(\theta) : C_n(\theta) \geq 1\}; \\ \theta_n^*(2) &\in \arg\max\{f_n(\theta) : C_n(\theta) \geq \sqrt{n}\}. \end{aligned}$$

The estimator $\theta_n^*(1)$ (proposed by Andradóttir [11]) chooses a solution with the highest estimated objective function value as the estimate of the optimal solution, while the estimator $\theta_n^*(2)$ (proposed by Andradóttir [14] for problems with countably infinite feasible regions) selects a solution with the highest estimated objective function value among a solutions that have been simulated at least $\sqrt{n}$ times (in the case the set of such solutions is empty,

$\theta_n^*(2) = \theta_n^*(1)$). Observe that the estimator $\theta_n^*(1)$ is aggressive with respect to faith in $f_n(\theta)$ for small $C_n(\theta)$, while $\theta_n^*(2)$ is more conservative in this respect.

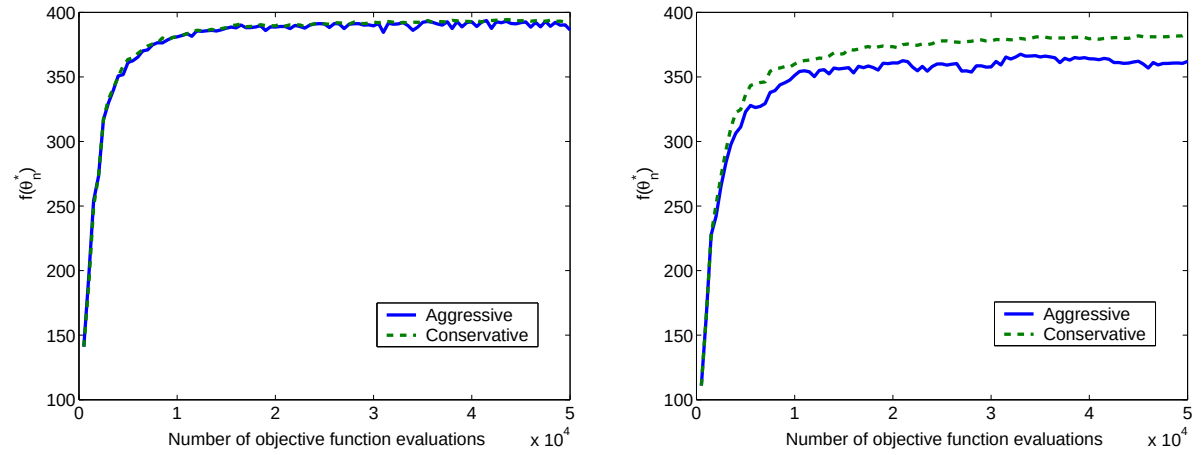
Numerical experiments were conducted to evaluate the performance of the optimization methods using the estimators $\theta_n^*(1)$ and $\theta_n^*(2)$ of the optimal solution. Observe that $\theta_n^*(2)$ is not relevant in the context of deterministic optimization. Consequently, the test problems considered for the purpose of comparing $\theta_n^*(1)$ and $\theta_n^*(2)$ are the unimodal problem with $\sigma^2 \in \{1,000, 160,000\}$ and the two hills problem with $\sigma^2 \in \{1, 50\}$ (the three stage buffer allocation problem is not used because the noise in the objective function estimates cannot be controlled). The simulation optimization algorithms employed in this experiment are R-BEESE, A-BEESE, and Global SA with parameter values provided in Table 3.2 (the Local SA method is not used due to its bad empirical performance on the problems, see Figure 3.4). The performance of each algorithm is averaged over 100 independent replications. Parts (a) and (b) of Figure 3.6 show the performance of the A-BEESE method with the estimators $\theta_n^*(1)$ and $\theta_n^*(2)$ (referred to as Aggressive and Conservative, respectively) on the unimodal problem with $\sigma^2 \in \{1,000, 160,000\}$. Similarly part (c) of Figure 3.6 shows the performance of Global SA with the two estimators on the unimodal problem with $\sigma^2 = 1,000$. The results depicted in parts (a) and (c) of Figure 3.6 are typical for all the algorithms under consideration when applied to problems with low noise, while the results in part (b) of Figure 3.6 are typical for the R-BEESE and A-BEESE methods when applied to solve problems with high noise.

From parts (a) and (b) of Figure 3.6, it is obvious that for the A-BEESE method, the estimator $\theta_n^*(1)$ performs better than the estimator $\theta_n^*(2)$ in the low noise setting and slightly worse in the high noise setting. Note that the average performance of the optimization algorithms with the estimator $\theta_n^*(2)$ is smoother than with the estimator $\theta_n^*(1)$ on problems with high noise. This is a consequence of the fact that $\theta_n^*(2)$ is more conservative than $\theta_n^*(1)$, i.e., it is more likely that a solution with high objective function estimate but low objective function value is chosen to be the estimate of the optimal solution by $\theta_n^*(1)$ than by $\theta_n^*(2)$. From parts (a) and (b) of Figure 3.6, it can be concluded that as the noise in the estimates of the objective function values increases, the estimator of the optimal solution



(a) A-BEESE method and $\sigma^2 = 1,000$

(c) Global SA and $\sigma^2 = 1,000$



(b) A-BEESE method and $\sigma^2 = 160,000$

(d) Global SA and $\sigma^2 = 160,000$

Figure 3.6: Comparison of estimators of the optimal solution on the unimodal problem with $\sigma^2 = 1,000$ and $\sigma^2 = 160,000$

should become more conservative. That is, a point with the highest estimated objective function value should not necessarily be picked as the estimate of the optimal solution if only a few observations have been collected at it and the observations of the objective function values are noisy.

In our experiments, the largest observed difference in the performance of the estimators $\theta_n^*(1)$ and $\theta_n^*(2)$ is when the Global SA algorithm is used to optimize the unimodal problem with $\sigma^2 = 160,000$. This result is depicted in part (d) of Figure 3.6, and a comparison with part (c) of Figure 3.4 shows that the use of a poor estimate of the optimal solution explains a substantial part of the poor performance of Global SA on this problem. The

improved performance of Global SA using the estimator $\theta_n^*(2)$ is due to the fact that the feasible space is large and the Global SA algorithm moves aggressively within the feasible space. This means that it is likely that a point with a low objective function value can have a high estimated objective function value for a long period of time before it is revisited by the algorithm. Such a situation is less likely to occur when features that aid estimation are incorporated into an algorithm (as is done in the R-BEESE and A-BEESE methods). Thus, as expected, larger benefits from the use of $\theta_n^*(2)$ as opposed to $\theta_n^*(1)$ can be obtained for simulation optimization algorithms that perform less estimation. Note however that the R-BEESE and A-BEESE algorithms still outperform Global SA when the estimator $\theta_n^*(2)$ is used. Possible reasons for this worse behavior can be that the algorithm is Markovian, it does not have exploitation components, and the underlying feasible region is large. Consequently, if a simulation optimization algorithm does not have good performance with the estimator $\theta_n^*(1)$, then it will not be necessarily the case that the use of $\theta_n^*(2)$ will improve the empirical performance of the method dramatically (i.e., if the algorithm is not "good," then improving the estimator of the optimal solution alone may not improve its performance drastically).

3.6 Conclusions

In this chapter, we have discussed desirable features that a simulation optimization algorithm should possess to have good empirical performance. In particular, our approach to solving simulation optimization problems involves maintaining an appropriate balance between exploration, exploitation, and estimation. The role and importance of each component is discussed and some guidelines are given on how to design effective random search methods within the proposed approach. Moreover, we have developed two new almost surely convergent random search methods possessing the desired features. These methods are intuitive, simple, flexible enough to allow an end-user to exploit the structure inherent in the optimization problem at hand, and also exhibit attractive empirical performance. Finally, we have demonstrated that although the estimator of the optimal solution proposed in Andradóttir [14] was originally designed for simulation optimization problems with countably infinite feasible regions, it also has good empirical performance on problems with finite

feasible regions and high noise in the estimates of the objective function values.

CHAPTER IV

AN AVERAGING FRAMEWORK FOR SIMULATION OPTIMIZATION WITH APPLICATIONS TO SIMULATED ANNEALING

4.1 Introduction

In this chapter, we present a framework for designing adaptive random search methods. Our framework is very general in that it outlines a broad class of methods intended for solving the simulation optimization problem (1.1). The methods are adaptive in that they use information gathered during previous iterations to decide on how simulation effort is expended in the current iteration. Also, our framework is based on averaging in that whenever estimates of the objective function values at some feasible solutions are required, these estimates are the averages of all observations collected at these solutions so far (as opposed to the averages of observations collected in the current iteration only). This may potentially lead to a significant reduction in the computational time required to solve the optimization problem (1.1) because the methods are not required to discard any information obtained during previous iterations of the algorithm. This feature is especially useful when estimating the performance measure of interest involves conducting a steady-state simulation because it allows us to continue simulations of sample paths that were started in previous iterations.

We use our framework to provide rigorous theoretic grounds for proving almost sure convergence of the sequence of estimates of the optimal solution to the set of global optima. In particular, methods within the framework are provably convergent under very mild assumptions. This feature allows practitioners and researchers to design numerically efficient random search methods for discrete simulation optimization that also can be easily shown to be theoretically convergent (by verifying the assumptions of our framework).

We also present a framework for point-based methods that is a special case of our general framework. Point-based methods include several random search algorithms discussed in

Chapter 2, like simulated annealing, stochastic ruler, stochastic comparison, etc. These search methods move iteratively from one feasible point to another based on some criteria. This special structure is useful in showing that point-based methods are convergent under less restrictive assumptions than the methods outlined by our general framework.

The algorithmic framework presented in this chapter is related to the frameworks in Andradóttir [13, 14]. However, the assumptions under which the algorithms within our framework are shown to converge are substantially different from those in Andradóttir [13, 14]. Some other works on frameworks for solving the simulation optimization problem (1.1) include Neddermeijer et al. [65], Ólafsson and Kim [68], and Boesel, Nelson, and Ishii [24].

The theoretical analysis of random search methods with averaging usually involves verifying that each feasible solution is sampled infinitely often with probability one in the limit. This is typically satisfied by incorporating some form of pure random search into a method (see, for instance, Pichitlamken and Nelson [71], Prudius and Andradóttir [73], Andradóttir [14], and Chapter 3 of this thesis), or it is assumed without providing justification (see Fox and Heine [31]). Incorporating a pure search component might not be desirable if it is difficult or even impossible to sample solutions from the entire feasible region using pure random search, or if the problem structure is best exploited using local neighborhoods. In contrast, our frameworks rely on assuming that when the random search method under consideration is applied to solve a deterministic version of the problem, it samples every feasible point infinitely often with probability one. Our assumption usually can be verified for a particular random search method without the need of incorporating a pure random search component into it, and usually boils down to analyzing a Markov chain that has been studied before (for the purpose of analysis of the original methods that did not involve averaging).

Also, in this chapter, we apply our frameworks to analyze the NP method of Pichitlamken and Nelson [71] and the SA algorithm with decreasing cooling schedule and (possibly) local neighborhoods. By virtue of doing so, we introduce two new and almost surely convergent variants of the SA algorithm with decreasing cooling schedule. Our two variants

of the SA algorithm only differ in the choice of the estimates of the objective function values at the current and candidate solutions used in each iteration to select the next current solution. In the first method, only observations obtained in the current iteration are used, while the second method utilizes all observations obtained so far at these two points (as in our framework).

The main contributions of this chapter are (i) flexible algorithmic frameworks that allow the design of adaptive (and hence time-inhomogeneous and non-Markovian) and almost surely convergent random search methods for discrete simulation optimization that use averaging and that do not require an artificial mechanism to ensure that each point is sampled infinitely often, (ii) two new and almost surely convergent variants of the SA algorithm with decreasing cooling schedule, (iii) enhanced theoretical and practical understanding of SA with decreasing cooling schedule for deterministic and stochastic optimization, and (iv) better practical understanding of the benefits of averaging in simulation optimization.

The remainder of this chapter is organized as follows. In Section 4.2, we present our algorithmic frameworks and also discuss the convergence of the algorithms within the frameworks. In Section 4.3, we demonstrate how our general framework can be used to prove the convergence of the NP method of Pichitlamken and Nelson [71]. In Section 4.4, we present our variants of the SA algorithm and discuss their convergence properties (the convergence of the second variant is based on our point-based framework). In Section 4.5, we provide some numerical results that investigate the effects of averaging, adaptiveness, and local versus global search for the proposed SA algorithms. Finally, concluding remarks are given in Section 4.6. A preliminary version of this chapter appeared in Prudius and Andradóttir [74].

4.2 Frameworks

In this section, we present, discuss, and analyze our algorithmic frameworks based on averaging for solving the optimization problem (1.1). In Section 4.2.1 we give our general framework for random search methods, while in Section 4.2.2 we provide our framework for point-based random search methods.

4.2.1 General Framework

In this section, we present and discuss a general algorithmic framework for random search methods with averaging. We also provide a convergence analysis for the algorithms within our framework and discuss the conditions under which the methods are guaranteed to converge. Now we present our framework.

Algorithm 4.1

Step 0: Let $n = 0$ and choose the initial sampling strategy $\mathcal{S}_n$. For all $\theta \in \Theta$, let $\Sigma_n(\theta) = 0$ and $C_n(\theta) = 0$.

Step 1: Generate a collection of solutions $\Theta_n \subset \Theta$ based on the sampling strategy $\mathcal{S}_n$, independent of the previous iterations.

Step 2: Given Θ_n, generate $K_n(\theta)$ additional observations $\{X_\theta^i\}_{i=C_n(\theta)+1}^{C_n(\theta)+K_n(\theta)}$ of X_θ for all $\theta \in \Theta_n$. For each $\theta \in \Theta_n$, let

$$\Sigma_{n+1}(\theta) = \Sigma_n(\theta) + \sum_{i=C_n(\theta)+1}^{C_n(\theta)+K_n(\theta)} h_\theta(X_\theta^i)$$

and $C_{n+1}(\theta) = C_n(\theta) + K_n(\theta)$. Moreover, let $C_{n+1}(\theta) = C_n(\theta)$ and $\Sigma_{n+1}(\theta) = \Sigma_n(\theta)$ for all $\theta \in \Theta \setminus \Theta_n$. Calculate $\hat{f}_{n+1}(\theta) = \Sigma_{n+1}(\theta)/C_{n+1}(\theta)$ for $\theta \in \Theta$ (use the convention $0/0 = -\infty$).

Step 3: Choose an updated sampling strategy $\mathcal{S}_{n+1}$ (see Assumption 4.7 below).

Step 4: Let $n = n+1$ and select an estimate of the optimal solution $\theta_n^* \in \arg\max_{\theta\in\Theta} \hat{f}_n(\theta)$. Go to Step 1.

We next comment on the algorithmic framework given above. Observe that the estimate of the objective function value at any solution is an average of all observations collected at this solution so far. However, our convergence results (with minor modifications) are also valid for other estimators of the objective function values (they hold with any strongly consistent estimators of the objective function values, and hence apply to both transient and steady-state performance measures, see Remark 4.1 at the end of this section). Moreover,

the algorithm above does not include a stopping criterion, which is consistent with the literature on random search methods because the convergence results are typically asymptotic in nature. Also, the number of feasible points sampled in Step 1 need not be specified for each iteration n in advance, but rather can be a parameter of the sampling strategy. Finally, the number of objective function observations $K_n(\theta)$ collected at a sampled solution θ in iteration n can depend on all the information gathered by the algorithm by iteration $n-1$. This extends previous work, where $K_n(\theta)$ is usually controlled deterministically (either kept fixed or required to grow deterministically with n), and hence this feature allows random search methods to be more adaptive to the information seen. It might, for instance, be desirable to keep $K_n(\theta)$ small in situations where a lot of observations have been collected at θ by iteration $n-1$, so that the variance of $\hat{f}_n(\theta)$ is small, and hence collecting more observations at θ might not produce a considerably better estimate of $f(\theta)$. The identification of good strategies for choosing $K_n(\theta)$ is beyond the scope of this dissertation.

From a computational standpoint, note that if the objective function estimates at some points are not used in updating the sampling strategy, then these estimates need not be calculated in Step 2. Similarly an estimate of the optimal solution need not be calculated at every iteration but can be simply calculated when the search is terminated.

In the subsequent analyses, "i.o." stands for "infinitely often" and $|A|$ denotes the cardinality of set A. We next present the convergence analysis of Algorithm 4.1. Before doing so we need to give the following assumptions and definitions.

Assumption 4.1. *The random elements used for estimating the objective function values are independent of the random elements used in the execution of algorithmic decisions (e.g., generating Θ_n and updating $\mathcal{S}_n$). Moreover, random elements involved in estimating the objective function values at different solutions are independent of each other and the random elements used in the execution of algorithmic decisions in different iterations are also independent of each other.*

Assumption 4.2. *Θ is a finite element set.*

Assumption 4.3. *For each $\theta \in \Theta$, there exists a set $V(\theta) = \{f_i(\theta) : i \in \mathbb{N}\} \subset \mathbb{R} \cup \{-\infty\}$*

such that the sequence $\{\hat{f}_n(\theta)\}_{n=1}^{\infty}$ *belongs to* $V(\theta)$*. Moreover, for each* $\theta \in \Theta$*, the sequence* $\{f_i(\theta)\}_{i\in\mathbb{N}}$ *satisfies the following condition* $\inf_{i,j\in\mathbb{N},i\neq j} |f_i(\theta) - f_j(\theta)| > 0$*.*

Assumption 4.4. *For each* $\theta \in \Theta$*,* $\sum_{i=1}^{n} h_\theta(X_\theta^i)/n$ *is a strongly consistent estimator of* $f(\theta)$*.*

Assumption 4.5. *For each* $\theta \in \Theta$*, the sequence* $\{K_n(\theta)\}$ *is such that*

$$\mathbb{P}\Big(\{\theta \in \Theta_n \text{ } i.o.\} \setminus \{C_n(\theta) \to \infty\}\Big) = 0.$$

Assumption 4.6. *For each* $k \in \mathbb{N}$*, the number of possible sampling strategies* $\mathcal{S}_k$ *is countable.*

Assumption 4.7. *For each* $n \in \mathbb{N}$*, the sampling strategy* $\mathcal{S}_{n+1}$ *can depend only on the objective function estimates* $\{\hat{f}_{n+1}(\theta)\}_{\theta\in\Theta}$*, the current sampling strategy* $\mathcal{S}_n$*, the iteration number* n*, and possibly other random elements that are independent of everything else.*

In Assumptions 4.1 and 4.7, by the term random elements, we mean standard uniform random numbers. Note that the updated sampling strategy $\mathcal{S}_{n+1}$ can depend deterministically on the objective function estimates $\{\hat{f}_{n+1}(\theta)\}_{\theta\in\Theta}$, but any random elements associated with updating the sampling strategy must be independent of these objective function estimates.

Let the class $\mathcal{C}$ of real-valued functions on Θ be defined as

$$\mathcal{C} = \left\{\tilde{f} : \Theta \to \mathbb{R} \middle| \text{for all } \theta \in \Theta, \exists i_\theta \text{ such that } \tilde{f}(\theta) = f_{i_\theta}(\theta)\right\}.$$

Also, let $Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta})$ denote Algorithm 4.1 applied to optimize the deterministic objective function $\tilde{f} \in \mathcal{C}$ initialized (in Step 0 of Algorithm 4.1) with iteration number $n = k$, an initial sampling distribution $\mathcal{S}_k$, $C_k(\theta)$ observations collected at each $\theta \in \Theta$, $\Sigma_k(\theta) = \tilde{f}(\theta) \times C_k(\theta)$ for all $\theta \in \Theta$, and $K_n(\theta) = 1$ for all $n \geq k$ and $\theta \in \Theta_n$. Note that the sample paths of $Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta})$ are well defined under Assumption 4.7.

Assumption 4.8. *For each* $\tilde{f} \in \mathcal{C}$*,* $k \in \mathbb{N}$*,* $\mathcal{S}_k$*, and* $C_k(\theta) \in \mathbb{N}$ *for all* $\theta \in \Theta$*,* $Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta})$ *samples each feasible solution infinitely often with probability one.*

We next discuss the assumptions under which the algorithms within our framework are guaranteed to converge. Assumption 4.1 imposes some conditions on the dependence structure of the random elements involved in Algorithm 4.1. These conditions are consistent with the random search literature, and are easy to satisfy because the random elements are under the control of a user. This assumption is used in the construction of the underlying sample space on which Algorithm 4.1 is defined, and is a crucial part of our proof.

Assumptions 4.2 through 4.4 are our structural assumptions concerning the underlying optimization problem. Assumption 4.2 is fairly standard for the discrete simulation optimization literature, with the exception of works by Andradóttir [14], Hong and Nelson [52], and Hong [50], where countably infinite feasible regions are considered. Assumption 4.3 states that an objective function estimate at any solution θ in any iteration can take at most countably many different values, and, moreover, there is always a minimal distance (or separation) between these possible values. We do not consider this assumption to be a serious restriction because most practical applications of Algorithm 4.1 are implemented on computers which have finite precision anyway. A similar assumption is used in Fox and Heine [31]. In Section 4.4.3 below, we provide discussion on how this assumption can be relaxed for the SA algorithm with averaging (see Algorithm 4.4 in Section 4.4.2). Assumption 4.4 also can be easily satisfied in practice, and holds, for instance, if for each $\theta \in \Theta$, $\{X_\theta^i\}_{i=1}^\infty$ are independent and identically distributed observations of X_θ (this follows from the Strong Law of Large Numbers). Notice that Assumptions 4.2 through 4.4 imply that there exists almost surely n_0 large enough such that $n \geq n_0$ implies that $\sum_{i=1}^n h_\theta(X_\theta^i)/n = f(\theta) \in V(\theta)$ for every $\theta \in \Theta$. This plays a crucial role in the proof of our main result concerning Algorithm 4.1.

Now we discuss the algorithmic Assumptions 4.5 through 4.8. Roughly speaking Assumption 4.5 says that the number of observations collected at any solution diverges to infinity provided that this solution is sampled infinitely often. Observe that Assumption 4.5 is satisfied if $K_n(\theta) = K \in \mathbb{N}^+$ for all $n \in \mathbb{N}$ and $\theta \in \Theta$. Assumption 4.6 is usually trivially satisfied by random search methods. For example, in point-based methods, the sampling strategy in iteration n depends only on the current point, and the set of current

points is finite under Assumption 4.2. Similarly, Assumption 4.7 is also usually trivially satisfied by random search methods. Indeed, all random search methods referenced in this thesis do satisfy this assumption. Consequently, Assumption 4.8 is usually the most difficult to verify. In this chapter, we provide two examples of independent interest involving the NP and SA optimization methods in which we show that this assumption is satisfied.

We now compare our framework to that of Andradóttir [13, 14]. Structurally, the primary difference between our frameworks is that the sampling strategy update step in our framework depends on the average of the objective function observations collected so far at each solution (this is not a requirement in Andradóttir [13, 14]). Secondly, we use a different estimator of the optimal solution. Both frameworks show that the sequence of estimators of the optimal solution converges almost surely to the set of globally optimal solutions. Despite this similarity, the assumptions under which they are convergent are quite different. The main assumption of our framework is Assumption 4.8 which states that the optimization method samples each feasible solution infinitely often with probability one when applied to solve a deterministic optimization problem, while the proof of the main result in Andradóttir [14] mostly relies on the assumptions that some optimal solution θ^* is sampled with probability $p > 0$ in each iteration independently of the prior activities of the search method and the objective function estimates at the nonoptimal points and θ^* behave probabilistically in a certain way with respect to $f(\theta^*)$.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space on which Algorithm 4.1 is defined. We now present our main result concerning Algorithm 4.1.

Theorem 4.1. *Suppose that Assumptions 4.1 through 4.8 are satisfied. Then the sequence* $\{\theta_n^*\}$ *generated by Algorithm* 4.1 *converges almost surely to the set* Θ^* *in the sense that for almost every* $\omega \in \Omega$*, there exists* $N(\omega)$ *such that* $n \geq N(\omega)$ *implies that* $\theta_n^*(\omega) \in \Theta^*$.

Proof: To prove the theorem, we will construct the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ in a specific manner. To save space and simplify notation, we will construct only the underlying sample space Ω. It should be obvious from the context what $\mathcal{F}$ and $\mathbb{P}$ are meant.

Observe that the random elements in Algorithm 4.1 are of two types, namely, the ones

needed for estimating the objective function value at each solution in Step 2 (defined on a sample space Ω_s) and the ones needed for execution of algorithmic decisions (defined on a sample space Ω_d). Hence, in view of Assumption 4.1, without loss of generality, we can assume that $\Omega = \Omega_d \times \Omega_s$. We will identify probability one subsets $\tilde{\Omega}_d \subset \Omega_d$ and $\tilde{\Omega}_s \subset \Omega_s$ that possess desirable properties. Then we will show that Algorithm 4.1 converges to the set Θ^* for almost every $\omega \in \tilde{\Omega}_d \times \tilde{\Omega}_s$.

We first construct the subset $\tilde{\Omega}_d \subset \Omega_d$. For each $n \in \mathbb{N}$, let Ω_n be a sample space on which the random elements for execution of algorithmic decisions in iteration n are defined, so that $\Omega_d = \prod_{n=0}^{\infty} \Omega_n$ (this is possible due to Assumption 4.1). Suppose that $Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta})$ uses the random elements defined on Ω_{k+n} at iteration n. Define $\Omega_k^{\infty} = \prod_{n=k}^{\infty} \Omega_n$. Observe that $Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta})$ is defined on the sample space Ω_k^{∞} because $\tilde{f}$ can be evaluated without noise.

For all $\tilde{f} \in \mathcal{C}$, $k \in \mathbb{N}$, $\mathcal{S}_k$, and $C_k(\theta) \in \mathbb{N}$ for all $\theta \in \Theta$, define

$$A_k(\tilde{f}, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta}) = \{\omega \in \Omega_k^{\infty} : Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta}) \text{ samples each } \theta \in \Theta \text{ i.o.}\}.$$

Then by Assumption 4.8 it follows that $\mathbb{P}(A_k(\tilde{f}, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta}))=1$. For each $k \in \mathbb{N}$, let

$$A_k = \bigcap_{\tilde{f}\in\mathcal{C}} \bigcap_{\mathcal{S}_k} \bigcap_{C_k(\theta)\in\mathbb{N}:\theta\in\Theta} A_k(\tilde{f}, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta}).$$

Assumptions 4.2, 4.3, and 4.6 ensure that the intersection above is taken over countably many sets. Thus, we have that $\mathbb{P}(A_k) = 1$ for all $k \in \mathbb{N}$. For each $k \in \mathbb{N}$, let $\tilde{A}_k = (\prod_{n=0}^{k-1} \Omega_n) \times A_k$. Obviously $\mathbb{P}(\tilde{A}_k) = 1$. Finally, let $\tilde{\Omega}_d = \bigcap_{k=0}^{\infty} \tilde{A}_k$, the set of all sample elements such that for all $\tilde{f} \in \mathcal{C}$, $k \in \mathbb{N}$, $\mathcal{S}_k$'s, and $C_k(\theta) \in \mathbb{N}$ for all $\theta \in \Theta$, we have that $Alg(\tilde{f}, k, \mathcal{S}_k, \{C_k(\theta)\}_{\theta\in\Theta})$ samples each feasible point infinitely often. Clearly, $\mathbb{P}(\tilde{\Omega}_d) = 1$.

Let $\tilde{\Omega}_s \subset \Omega_s$ be such that $\sum_{i=1}^{n} h_\theta(X_\theta^i)/n \to f(\theta)$ as $n \to \infty$ for all $\theta \in \Theta$, where $\{X_\theta^i\}_{i=1}^{\infty}$ are observations of X_θ. Then Assumptions 4.2 and 4.4 imply that $\mathbb{P}(\tilde{\Omega}_s) = 1$.

For each $\theta \in \Theta$, let Ω_θ be the null set of Assumption 4.5. Let $\bar{\Omega} = \Omega \setminus \cup_{\theta\in\Theta}\Omega_\theta$. Note that for each $\theta \in \Theta$, if $\omega \in \bar{\Omega}$ and $\theta \in \Theta_n(\omega)$ infinitely often, then $C_n(\theta, \omega) \to \infty$ as $n \to \infty$. Assumptions 4.2 and 4.5 ensure that $\mathbb{P}(\bar{\Omega}) = 1$.

Fix $\omega \in (\tilde{\Omega}_d \times \tilde{\Omega}_s) \cap \bar{\Omega}$. We next show that Algorithm 4.1 samples each feasible solution

infinitely often under this ω. We proceed by contradiction. Let

$$\bar{\Theta}(\omega) = \{\theta \in \Theta : \text{ Algorithm 4.1 samples } \theta \text{ i.o. under } \omega\}.$$

Suppose that $\bar{\Theta}(\omega) \neq \Theta$. Then by Assumptions 4.2 and 4.3 and the choice of ω, there exists an iteration number $n_0(\omega) \in \mathbb{N}$ such that $n \geq n_0$ implies that $\Theta_n \subset \bar{\Theta}(\omega)$ and $\hat{f}_n(\theta) = f(\theta)$ for all $\theta \in \bar{\Theta}(\omega)$. Consequently, the objective function estimates from this point on do not change. Denote this objective function estimate by $\tilde{f}(\omega)$. Assumption 4.3 ensures that $\tilde{f}(\omega) \in \mathcal{C}$ and Assumption 4.7 ensures that Algorithm 4.1 couples with $Alg(\tilde{f}(\omega), n_0, \mathcal{S}_{n_0}, \{C_{n_0}(\theta)\}_{\theta\in\Theta})$ from iteration n_0 (that is, the sets Θ_n of points sampled by Algorithm 4.1 and $Alg(\tilde{f}(\omega), n_0, \mathcal{S}_{n_0}, \{C_{n_0}(\theta)\}_{\theta\in\Theta})$ coincide for all $n \geq n_0$). But by the choice of ω, we know that $Alg(\tilde{f}(\omega), n_0, \mathcal{S}_{n_0}, \{C_{n_0}(\theta)\}_{\theta\in\Theta})$ samples each feasible point infinitely often. This provides a contradiction, and hence we have shown that Algorithm 4.1 samples all $\theta \in \Theta$ infinitely often under ω.

Then by Assumptions 4.2 and 4.3 and the choice of ω, there exists an $N(\omega) \in \mathbb{N}$ such that for all $\theta \in \Theta$ and $n \geq N(\omega)$, we have that $\hat{f}_n(\theta) = f(\theta)$. This shows that $\theta_n^* \in \Theta^*$ for all $n \geq N(\omega)$. ■

The next remark provides another estimator of the objective function values with which Algorithm 4.1 is also convergent. Such an estimator is likely to occur when Algorithm 4.1 is applied to solve optimization problems with a steady-state performance measure.

Remark 4.1. Let the estimate of the objective function value at $\theta \in \Theta$ after n iterations be

$$\hat{f}_{n+1}(\theta) = \frac{1}{C_{n+1}(\theta)} \int_0^{C_{n+1}(\theta)} h'_\theta(X_\theta(t))\,\mathrm{d}t,$$

provided that $C_{n+1}(\theta) > 0$, and $\hat{f}_n(\theta) = -\infty$ otherwise, where h'_θ is some deterministic function and $X_\theta = \{X_\theta(t) : t \geq 0\}$ is a stochastic process. Note that in this case $C_n(\theta)$ is the length of the simulation run conducted at θ by the beginning of iteration n and $K_n(\theta)$ is the additional length of time for which X_θ is simulated in iteration n (i.e., from time $C_n(\theta)$ to $C_{n+1}(\theta)$). The interested reader is referred to Section 3.3.2 of this dissertation for a discussion on what this way of simulating sample paths entails. In this case, instead of Assumption 4.4, we assume that $f_T(\theta) = \int_0^T h'_\theta(X_\theta(t))\,\mathrm{d}t/T$ is a strongly consistent

estimator of $f(\theta)$ for all $\theta \in \Theta$. Then, under the conditions of Theorem 4.1, Algorithm 4.1 with such estimators of the objective function values is also convergent with probability one.

4.2.2 Framework for Point-Based Methods

In this section we present a special case of the framework of Section 4.2.1 that is designed to analyze point-based random search methods. We will show that a class of point-based methods is almost surely convergent under less restrictive assumptions than the general random search method outlined in Algorithm 4.1. The outline of the point-based methods we consider is given in Algorithm 4.2.

Algorithm 4.2

Step 0: Let $n = 0$ and choose a starting point $\theta_n \in \Theta$. For all $\theta \in \Theta$, let $\Sigma_n(\theta) = 0$ and $C_n(\theta) = 0$.

Step 1: Given $\theta_n = \eta$, generate a candidate solution $\theta'_n \in \Theta$ independent of everything else such that $\mathbb{P}[\theta'_n = \eta' | \theta_n = \eta] = Q_n(\eta, \eta')$ for all $\eta' \in \Theta$. Let $j = 1$.

Step 2: Given $\theta_n = \eta$ and $\theta'_n = \eta'$, generate $K_n^j(\eta)$ and $K_n^j(\eta')$ additional observations $\{X_\eta^i\}_{i=C_n(\eta)+1}^{C_n(\eta)+K_n^j(\eta)}$ and $\{X_{\eta'}^i\}_{i=C_n(\eta')+1}^{C_n(\eta')+K_n^j(\eta')}$ of X_η and $X_{\eta'}$, respectively. For $\theta = \eta, \eta'$, let

$$\Sigma_{n+1}(\theta) = \Sigma_n(\theta) + \sum_{i=C_n(\theta)+1}^{C_n(\theta)+K_n^j(\theta)} h_\theta(X_\theta^i)$$

and $C_{n+1}(\theta) = C_n(\theta) + K_n^j(\theta)$. Moreover, let $C_{n+1}(\theta) = C_n(\theta)$ and $\Sigma_{n+1}(\theta) = \Sigma_n(\theta)$ for all $\theta \in \Theta$, $\theta \neq \eta, \eta'$. Calculate $\hat{f}_{n+1}(\theta) = \Sigma_{n+1}(\theta)/C_{n+1}(\theta)$ for $\theta \in \Theta$ (use the convention $0/0 = -\infty$).

Step 3: Given $\theta_n = \eta$ and $\theta'_n = \eta'$, determine if more observations need to be collected at η and η' using the estimates $\hat{f}_{n+1}(\eta)$ and $\hat{f}_{n+1}(\eta')$. If so, let $j = j + 1$ and $C_n(\theta) = C_{n+1}(\theta)$ and $\Sigma_n(\theta) = \Sigma_{n+1}(\theta)$ for $\theta = \eta, \eta'$, and go to Step 2. Else determine the next point $\theta_{n+1} \in \{\theta_n, \theta'_n\}$ (see Assumption 4.9 below), and go to Step 4.

Step 4: Let $n = n+1$ and select an estimate of the optimal solution $\theta_n^* \in \arg\max_{\theta \in \Theta} \hat{f}_n(\theta)$. Go to Step 1.

Note that Algorithm 4.2 is a special case of Algorithm 4.1. We now briefly comment on point-based methods. These methods iteratively move from one feasible point to another. The sampling strategy in iteration n is completely determined by the current point θ_n and the neighbor generation probability matrix Q_n that is specified in advance of executing an algorithm. A single candidate solution θ'_n is generated in Step 1 based on the sampling distribution $Q_n(\theta_n, \cdot)$, and the set Θ_n of sampled points in iteration n is $\{\theta_n, \theta'_n\}$. Finally, the sampling strategy in Step 3 is updated by selecting the next current iterate θ_{n+1}, so that the sampling distribution in the next iteration becomes $Q_{n+1}(\theta_{n+1}, \cdot)$. As before, the number of observations collected at the current and candidate solutions can depend on all the information obtained by the algorithm so far and possibly on the realization of some other random elements that are independent of everything else. Point-based methods differ primarily in the way the next current point θ_{n+1} is selected. We next state conditions that replace Assumptions 4.7 and 4.8 in the case of point-based methods.

Assumption 4.9. *For each $n \in \mathbb{N}$, the next current iterate θ_{n+1} is chosen from $\Theta_n = \{\theta_n, \theta'_n\}$ based only on $\hat{f}_{n+1}(\theta_n)$, $\hat{f}_{n+1}(\theta'_n)$, θ_n, n, and possibly other random elements that are independent of everything else.*

For each $\theta \in \Theta$ and $k \in \mathbb{N}$, let $Alg(\theta, k)$ denote Algorithm 4.2 for optimizing the deterministic objective function f initialized (in Step 0 of Algorithm 4.2) with the starting point θ and iteration number $n = k$, with the rest of initialization being as before, and $K_n(\theta) = 1$ for all $n \geq k$ and $\theta \in \Theta_n$. The sample paths of $Alg(\theta, k)$ are well defined under Assumption 4.9.

Assumption 4.10. *For each $\theta \in \Theta$ and $k \in \mathbb{N}$, $Alg(\theta, k)$ samples each feasible point infinitely often with probability one.*

We next present our convergence result concerning the class of point-based methods outlined in Algorithm 4.2.

Theorem 4.2. *Suppose that Assumptions 4.1 through 4.5, 4.9, and 4.10 are satisfied. Then the sequence $\{\theta^*_n\}$ generated by Algorithm* 4.2 *converges almost surely to the set Θ^* in the*

sense that for almost every $\omega \in \Omega$*, there exists* $N(\omega)$ *such that* $n \geq N(\omega)$ *implies that* $\theta_n^*(\omega) \in \Theta^*$.

Proof: The proof of this result is similar to the proof of Theorem 4.1 (recall that Assumption 4.6 is trivially satisfied by point-based methods as long as Assumption 4.2 holds, see Section 4.2.1). Again, without loss of generality, we can assume that $\Omega = \Omega_d \times \Omega_s$. We define $\tilde{\Omega}_s$ and $\bar{\Omega}$ as in the proof of Theorem 4.1, while $\tilde{\Omega}_d$ is constructed similarly except for the following two modifications: (i) $A_k(\cdot)$ is substituted by

$$A_k(\theta) = \{\omega \in \Omega_k^\infty : Alg(\theta, k) \text{ samples each } \theta' \in \Theta \text{ i.o.}\}$$

and (ii) A_k is defined as $A_k = \cap_{\theta \in \Theta} A_k(\theta)$. Note that by Assumption 4.10, we have again that $\mathbb{P}(A_k(\theta)) = 1$ for all $k \in \mathbb{N}$ and $\theta \in \Theta$.

The coupling argument used to show that every feasible point is sampled infinitely often with probability one is slightly different in this case. Fix $\omega \in (\tilde{\Omega}_d \times \tilde{\Omega}_s) \cap \bar{\Omega}$ and let

$$\bar{\Theta}(\omega) = \{\theta \in \Theta : \text{ Algorithm 4.2 samples } \theta \text{ i.o. under } \omega\}.$$

Suppose that $\bar{\Theta}(\omega) \neq \Theta$. Then by Assumptions 4.2 and 4.3 and the choice of ω, there exists $n_0(\omega) \in \mathbb{N}$ such that $n \geq n_0$ implies that $\Theta_n \subset \bar{\Theta}(\omega)$ and $\hat{f}_n(\theta) = f(\theta)$ for all $\theta \in \bar{\Theta}(\omega)$. Hence, by Assumption 4.9, Algorithm 4.2 couples with $Alg(\theta_{n_0}, n_0)$ from iteration $n_0(\omega)$. But by the choice of ω, we know that $Alg(\theta_{n_0}, n_0)$ samples each feasible point infinitely often. This provides a contradiction, and hence we have shown that Algorithm 4.2 samples all $\theta \in \Theta$ infinitely often under ω. The rest of the proof is the same as that of Theorem 4.1.

■

Remark 4.2. From the proofs of Theorems 4.1 and 4.2, it is clear that Algorithms 4.1 and 4.2 visit each alternative infinitely often with probability one.

Remark 4.3. The conclusion of Remark 4.1 is also valid for Algorithm 4.2, provided that the conditions of Theorem 4.2 hold.

We next discuss the assumptions under which Theorem 4.2 holds. The only significant difference between Theorems 4.1 and 4.2 are Assumptions 4.8 and 4.10. Note that Assumption 4.10 is less restrictive than Assumption 4.8 because it requires a point-based method

to sample each solution infinitely often with probability one for the single deterministic objective function f, as opposed to for the entire class of functions $\mathcal{C}$ in the case of a general random search method. The reason we are able to prove that point-based methods converge under a less restrictive assumption than a general random search method is that the updated sampling strategy in Step 3 of Algorithm 4.1 for point-based methods only depends on "local" information (information about the current and candidate solutions), while in general random search method it can depend on "global" information (information about every feasible point).

4.3 Convergence of the Nested Partitions Method

In this section we provide an example illustrating how our convergence framework and Theorem 4.1 can be applied to prove the almost sure convergence of the Nested Partitions (NP) method of Pichitlamken and Nelson [71]. In their notation, the result of this section is applicable to the NP and NP+SSM methods. The convergence of these methods has been proven by other methods in Pichitlamken and Nelson [71].

The basic idea of the method is to iteratively partition the feasible region and to spend more simulation effort in the subregion that contains the solution with the highest estimated objective function value (called the most-promising region). Observe that the current most-promising region and the partitioning scheme in the NP method completely determine the sampling strategy. Because the partitioning scheme is kept fixed throughout the search, the sampling strategy only depends on the current most-promising region. We first identify how each step in the framework corresponds to the steps of the NP method. Step 0 of the framework corresponds to the *Initialization* step in Pichitlamken and Nelson [71] with the initial sampling strategy $\mathcal{S}_0$ being determined by the initial most-promising region Θ. Step 1 corresponds to *Partitioning* and *Sampling*. Step 2 represents *Selection of the Best Solution* with or without the use of the SSM procedure. Step 3 corresponds to *Updating the Most-Promising Region* and *Restart*, while Step 4 represents *Search Termination*. We have the following corollary of Theorem 4.1 for NP.

Corollary 4.1. Suppose that Assumptions 4.2 through 4.5 are satisfied. Then, the conclusion

of Theorem 4.1 holds for the NP method.

Proof: First, note that Assumption 4.1 is satisfied by the NP method. Assumption 4.2 and the fact that the sampling strategy is determined by the current most-promising region ensure that Assumption 4.6 holds. Assumption 4.7 is also trivially satisfied. Consequently, it only remains to show that Assumption 4.8 is satisfied.

Fix $\tilde{f} \in \mathcal{C}$, $k \in \mathbb{N}$, $C_k(\theta) \in \mathbb{N}$ for all $\theta \in \Theta$, and the current most-promising region $\mathcal{S}_k$. Also fix $\theta \in \Theta$. Note that $\mathbb{P}(\theta \in \Theta_n | \mathcal{S}_n = R) \geq 1/|\Theta|$ for all $n \geq k$ and $R \in \mathcal{R}$, where $\mathcal{R}$ denotes all possible subsets of Θ. For all $n \geq k$, let $A_n = \{\omega \in \Omega : \theta \in \Theta_n(\omega)\}$ and $\mathcal{G}_n = \sigma\{A_k, \ldots, A_n\}$, the σ-algebra generated by $A_k, \ldots, A_n$. Moreover, for $n \geq k$, define $\mathcal{A}_n = \{A = \cap_{j=k}^{n} B_j : B_j \in D_j \text{ for all } j = k, \ldots, n\}$, where $D_j = \{A_j, (A_j)^c\}$. Note that any event $A \in \mathcal{A}_n$ indicates in what iterations (from k to n) a solution θ is sampled. Let I_A be the indicator function of a set A. Then, for all $n \geq k+1$, almost surely we have that

$$\begin{aligned}
\mathbb{P}(\theta \in \Theta_n | \mathcal{G}_{n-1}) &= \sum_{A \in \mathcal{A}_{n-1}} \mathbb{P}(\theta \in \Theta_n | A) I_A \\
&= \sum_{A \in \mathcal{A}_{n-1}} \sum_{R \in \mathcal{R}} \mathbb{P}(\theta \in \Theta_n | A, \mathcal{S}_n = R) \mathbb{P}(\mathcal{S}_n = R | A) I_A \\
&= \sum_{A \in \mathcal{A}_{n-1}} \sum_{R \in \mathcal{R}} \mathbb{P}(\theta \in \Theta_n | \mathcal{S}_n = R) \mathbb{P}(\mathcal{S}_n = R | A) I_A \\
&\geq 1/|\Theta| \sum_{A \in \mathcal{A}_{n-1}} \sum_{R \in \mathcal{R}} \mathbb{P}(\mathcal{S}_n = R | A) I_A \\
&= 1/|\Theta| \sum_{A \in \mathcal{A}_{n-1}} I_A = 1/|\Theta|. \qquad (4.1)
\end{aligned}$$

The third equality follows from the observation that the probability of sampling θ in iteration n depends on the current most-promising region and does not depend further on whether θ has been sampled or not in the previous iterations.

Equation (4.1) yields that

$$\sum_{n=k+1}^{\infty} \mathbb{P}(\theta \in \Theta_n | \mathcal{G}_{n-1}) \geq \sum_{n=k+1}^{\infty} 1/|\Theta| = +\infty$$

almost surely. By the conditional Borel-Cantelli lemma (see, e.g., Corollary 2.3 on page 32 in Hall and Heyde [43]) we conclude that the NP method samples θ infinitely often with probability one. This and Assumption 4.2 ensure that the NP method samples every

feasible solution infinitely often with probability one. Hence, Assumption 4.8 is satisfied by the NP method. The result of the corollary now follows from Theorem 4.1. ■

Remark 4.4. Corollary 4.1 extends the NP method of Pichitlamken and Nelson [71] in that the number of observations collected at the sampled point can be more adaptive to the information gathered by the algorithm as opposed to being bounded from below by a positive constant (implying that Assumption 4.5 holds). Moreover, we do not require the objective function observations at each feasible point be independent and identically distributed as long as they satisfy Assumption 4.4. This comes at a cost of Assumption 4.3 that we consider not being restrictive due to practical considerations (see Section 4.2.1).

4.4 Convergence of New Variants of the Simulated Annealing Algorithm

In this section we present two new variants of the SA algorithm and discuss their convergence properties. In Section 4.4.1 we present our first variant of SA that does not employ averaging, while in Section 4.4.2 we give our second variant that is within our averaging framework given by Algorithm 4.2. In Section 4.4.3 we discuss how certain assumptions needed for the convergence of our second variant can be relaxed.

4.4.1 Simulated Annealing without Averaging

In this section, we present our first variant of the SA algorithm and also state and prove its convergence properties. This algorithm is *not* within our frameworks because it does not involve averaging. We present this method because it is of independent interest, and also because its convergence analysis is used to prove the convergence of our second variant of the SA algorithm, that does fall within our framework.

The search method considered in this section uses a decreasing cooling schedule $\{T_n\}$, and the number of observations of the objective function values taken at the current and candidate solutions in each iteration is equal to a constant K. As the estimator of the optimal solution, we use the state that has the highest estimated objective function value.

Now we are ready to state our variant of the SA algorithm. For each $n \in \mathbb{N}$, θ_n is the current solution, θ'_n is the candidate solution, and θ^*_n is the estimator of the optimal solution

in iteration n. Finally, $[x]^+ = \max\{x, 0\}$ for all $x \in \mathbb{R}$.

Algorithm 4.3

Step 0: Identical to Step 0 of Algorithm 4.2.

Step 1: Identical to Step 1 of Algorithm 4.2.

Step 2: Given $\theta_n = \eta$ and $\theta'_n = \eta'$, generate observations $\{X_\eta^i\}_{i=C_n(\eta)+1}^{C_n(\eta)+K}$ of X_η and $\{X_{\eta'}^i\}_{i=C_n(\eta')+1}^{C_n(\eta')+K}$ of $X_{\eta'}$. For $\theta = \eta, \eta'$, let

$$\Sigma_{n+1}(\theta) = \Sigma_n(\theta) + \sum_{i=C_n(\theta)}^{C_n(\theta)+K} h_\theta(X_\theta^i)$$

and $C_{n+1}(\theta) = C_n(\theta) + K$. Moreover, let $C_{n+1}(\theta) = C_n(\theta)$ and $\Sigma_{n+1}(\theta) = \Sigma_n(\theta)$ for all $\theta \in \Theta$, $\theta \neq \eta, \eta'$. Calculate $\hat{f}_{n+1}(\theta) = \sum_{i=C_n(\theta)+1}^{C_{n+1}(\theta)} h_\theta(X_\theta^i)/K$ for $\theta = \eta, \eta'$.

Step 3: Given $\theta_n = \eta$ and $\theta'_n = \eta'$, generate $U_n \sim U[0, 1]$ (independently of all other random elements) and set

$$\theta_{n+1} = \begin{cases} \theta'_n & \text{if } U_n \leq G_n(\eta, \eta'), \\ \theta_n & \text{otherwise,} \end{cases}$$

where

$$G_n(\eta, \eta') = \exp\left[\frac{-[\hat{f}_{n+1}(\eta) - \hat{f}_{n+1}(\eta')]^+}{T_n}\right].$$

Step 4: Identical to Step 4 of Algorithm 4.2.

We next discuss the relationship of this algorithm to SA algorithms for stochastic optimization available in the literature. Algorithm 4.3 resembles the methods of Gelfand and Mitter [36], Fox and Heine [31], and Gutjahr and Pflug [41] in that it is an SA algorithm with a decreasing cooling schedule, as opposed to having a constant temperature like the algorithms in Alrefaei and Andradóttir [3]. On the other hand, Algorithm 4.3 resembles Algorithm 2 in Alrefaei and Andradóttir [3] in that it uses the state with the highest estimated objective function value as the estimator of the optimal solution, while Gelfand and Mitter [36], Fox and Heine [31], and Gutjahr and Pflug [41] use the current solution to estimate the optimal solution. Also, we do not require the number of observations collected at the

current and candidate solutions considered in a particular iteration to increase at a specific rate as the iteration number grows, as is required in the method of Gelfand and Mitter [36] and Gutjahr and Pflug [41]. We instead keep the number of observations collected at the current and candidate solutions per iteration constant throughout the search (similar to Algorithm 2 of Alrefaei and Andradóttir [3]). Hence, our variant requires less computation time per iteration as the number of iterations becomes large. Our method does not employ averaging unlike the SA algorithm of Fox and Heine [31]. Also, we allow the probability distribution Q_n that controls how a candidate solution is generated in the neighborhood of a current solution to depend deterministically on the iteration number, while it is assumed to be constant in all the other aforementioned SA algorithms.

Now we discuss the convergence properties of Algorithm 4.3. Suppose that all random elements in Algorithm 4.3 (ones needed for generating candidate solutions and simulating their performance, plus the U_n's) are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We show that the estimator θ_n^* converges to the set of global optimal solutions Θ^* for almost every $\omega \in \Omega$. To prove this we adopt an approach similar to that of Mitra, Romeo, and Sangiovanni-Vincentelli [63] who have analyzed the SA algorithm for deterministic optimization. In particular, they show that the sequence of current iterates generated by SA converges in probability to the set Θ^*, while we show that the sequence of estimates of the optimal solution converges almost surely to the set Θ^* even when the objective function is stochastic. Before proceeding to the proof, we need to give the following assumptions and definitions.

Assumption 4.11. *We assume that $Q_n \to Q$ elementwise as $n \to \infty$, where Q is an $|\Theta| \times |\Theta|$ transition matrix of an irreducible Markov chain (MC).*

Assumption 4.12. *The deterministic cooling schedule $\{T_n\}$ is such that $T_{n+1} \leq T_n$ for all $n \in \mathbb{N}$ and $\lim_{n\to\infty} T_n = 0$.*

Assumption 4.13. *For each $\theta \in \Theta$, $\{X_\theta^i\}_{i=1}^\infty$ are independent and identically distributed random elements with the law of X_θ.*

For each $\theta \in \Theta$, let $N(\theta) = \{\theta' \in \Theta : Q(\theta, \theta') > 0\}$ be the set of limiting neighbors

of θ. Let Θ_L be the set of local minima for the objective function f with respect to the neighborhood graph G induced by N; i.e.,

$$\Theta_L = \left\{\theta \in \Theta : f(\theta) \leq f(\theta'), \forall \theta' \in N(\theta)\right\}. \tag{4.2}$$

Note that under Assumption 4.11, the condition $\Theta_L = \Theta$ implies that f is constant on the feasible space Θ. Thus, without loss of generality, we can impose the following assumption.

Assumption 4.14. *Θ_L is a proper subset of Θ.*

For each $\theta \in \Theta$, let $\hat{f}(\theta) = \sum_{i=1}^{K} h_\theta(X_\theta^i)/K$. Then define the maximum relative depth of the objective function in the neighborhood graph G as

$$L = \max_{\theta \in \Theta} \max_{\theta' \in N(\theta)} \mathbb{E}\left[[\hat{f}(\theta) - \hat{f}(\theta')]^+\right]. \tag{4.3}$$

Let

$$r = \max_{\theta \in \Theta} \max_{\theta' \in \Theta} d(\theta, \theta'), \tag{4.4}$$

where $d(\theta, \theta')$ is the distance of θ' from θ measured by the length (number of edges) of the minimum length path from θ to θ' in G subject to the condition that the path contain at least one point in $\Theta \setminus \Theta_L \neq \emptyset$. Note that if r' is defined as r with the exception that the minimum length path need not contain a point in $\Theta \setminus \Theta_L$ (so that r' can be viewed as a true diameter of the graph G), then $r \leq 2r'$. Note that under Assumption 4.14, $r \geq 2$. Also, let

$$q = \frac{1}{2} \min_{\theta \in \Theta} \min_{\theta' \in N(\theta)} Q(\theta, \theta').$$

Note that under Assumption 4.2, $q > 0$. The proofs of all lemmas in this section are given in Appendix A. The next lemma shows that $L > 0$.

Lemma 4.1. *Suppose that Assumptions 4.11 and 4.14 are satisfied. Then $L > 0$.*

For $n \in \mathbb{N}$ and $\theta \in \Theta$, define $N_n(\theta) = \{\theta' \in \Theta : Q_n(\theta, \theta') > 0\}$. Under Assumptions 4.1 and 4.13, the stochastic process $W = \{\theta_n\}$ generated by Algorithm 4.3 is a discrete-time nonhomogeneous Markov chain with transition probability matrices $\mathbf{P}_n$ given by

$$\mathbf{P}_n(\theta, \theta') = \begin{cases} Q_n(\theta, \theta')\mathbb{E}[\exp(-[\hat{f}(\theta) - \hat{f}(\theta')]^+/T_n] & \text{if } \theta' \in N_n(\theta), \\ 1 - \sum_{\theta' \in N_n(\theta)} \mathbf{P}_n(\theta, \theta') & \text{if } \theta' = \theta, \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Define the m-step transition matrix $\mathbf{P}_{n,n+m} = \prod_{i=0}^{m-1} \mathbf{P}_{n+i}$. Next we derive a lower bound on the value of each entry in the matrix $\mathbf{P}(n, n+r)$ for sufficiently large values of n.

Lemma 4.2. *Suppose that Assumptions 4.1, 4.2, 4.11, 4.12, 4.13, and 4.14 hold. Then there exists $n_1 \in \mathbb{N}$ such that for all $\theta, \theta' \in \Theta$ and $n \geq n_1 r$, we have that $\mathbf{P}_{n-r,n}(\theta, \theta') \geq q^r \exp(-rL/T_{n-1})$.*

We need the following technical lemma.

Lemma 4.3. *Suppose that Assumption 4.12 holds. Then the following are equivalent:*

$$\begin{aligned} &(i) \quad \textstyle\sum_{n=1}^{\infty} \exp\left(-\frac{rL}{T_{nr+k-1}}\right) = +\infty \text{ for all } k \in \mathbb{N}, \\ &(ii) \qquad \textstyle\sum_{n=0}^{\infty} \exp\left(-\frac{rL}{T_n}\right) = +\infty. \end{aligned} \tag{4.6}$$

For $\theta \in \Theta$ and $n \in \mathbb{N}$, let $A_\theta^n = \{\omega \in \Omega : \theta_{nr}(\omega) = \theta\}$. Also for each $n \in \mathbb{N}$, let $\mathcal{F}_n$ be the σ-algebra generated by $\{\theta_{jr}\}_{j=0}^n$. Observe that $A_\theta^n \in \mathcal{F}_n$. The next proposition provides a sufficient condition on the cooling schedule which ensures that for each $\theta \in \Theta$, A_θ^n occurs i.o. with probability one.

Proposition 4.1. Suppose that Assumptions 4.1, 4.2, 4.11, 4.12, 4.13, and 4.14 are satisfied. Then, for each $\theta \in \Theta$, $\mathbb{P}(A_\theta^n \text{ i.o.}) = 1$ provided that the cooling schedule satisfies equation (4.6).

Proof: Fix $\theta \in \Theta$. Then we have that

$$\begin{aligned} \sum_{n=1}^{\infty} \mathbb{P}(A_\theta^n | \mathcal{F}_{n-1}) &\geq \sum_{n=n_1}^{\infty} \mathbb{P}(A_\theta^n | \mathcal{F}_{n-1}) = \sum_{n=n_1}^{\infty} \mathbb{P}(\theta_{nr} = \theta | \theta_{(n-1)r}) \\ &\geq \sum_{n=n_1}^{\infty} q^r \exp(-rL/T_{nr-1}) = +\infty. \end{aligned}$$

The first equality follows from the fact that W is a Markov chain (see equation (4.5)) and from the definition of the event A_θ^n. The second inequality follows by Lemma 4.2. The final equality follows from Lemma 4.3 and (4.6). The conditional Borel-Cantelli lemma (see, e.g., Corollary 2.3 on page 32 in Hall and Heyde [43]) now implies that $\mathbb{P}(A_\theta^n \text{ i.o.}) = 1$. ■

The result in Proposition 4.1 is also of interest in the context of deterministic optimization. In particular, suppose that the sequence $\{\theta_n\}$ generated by the SA algorithm for

deterministic optimization (i.e., Algorithm 4.3 with $K = 1$) converges in probability to the set Θ^* (for conditions under which this happens see Hajek [42] and Tsitsiklis [88]). These two results together imply that as n gets large, the sequence $\{\theta_n\}$ tends to spend more time at "good" solutions, but still it visits every solution infinitely often.

Next we state and prove our main convergence result for Algorithm 4.3.

Theorem 4.3. *Suppose that Assumptions 4.1, 4.2, 4.11, 4.12, and 4.13 are satisfied and the cooling schedule satisfies equation (4.6), where L is defined in equation (4.3). Then the sequence $\{\theta_n^*\}$ generated by Algorithm* 4.3 *converges almost surely to the set Θ^* in the sense that for almost every $\omega \in \Omega$, there exists $N(\omega)$ such that $n \geq N(\omega)$ implies that $\theta_n^*(\omega) \in \Theta^*$.*

Proof: Observe that without loss of generality, we can assume that Assumption 4.14 holds. Let $\epsilon = \max_{\theta \in \Theta} f(\theta) - \max_{\theta \in \Theta \setminus \Theta^*} f(x) > 0$ under Assumptions 4.2 and 4.14. From Proposition 4.1 it follows that $C_n(\theta) \to \infty$ almost surely for every $\theta \in \Theta$. Then, the Strong Law of Large Numbers and Assumption 4.13 imply that $\Sigma_n(\theta)/C_n(\theta) \to f(\theta)$ almost surely as $n \to \infty$ for all $\theta \in \Theta$ (denote this set of realizations by $\tilde{\Omega}$). Fix $\omega \in \tilde{\Omega}$. Thus, by Assumption 4.2 there exists $N(\omega) \in \mathbb{N}$ such that $|\Sigma_n(\theta, \omega)/C_n(\theta, \omega) - f(\theta)| < \epsilon/2$ for all $n \geq N(\omega)$ and $\theta \in \Theta$. This implies that $\theta_n^*(\omega) \in \Theta^*$ for $n \geq N(\omega)$ and hence the proof is complete. ■

Remark 4.5. For each $n \in \mathbb{N}$, let $T_n = C/\log(n + k)$. Then this cooling schedule satisfies Assumption 4.12 and equation (4.6) provided that $C \geq rL$ and $k > 1$. Moreover, from the proofs in this section, it should be clear that Theorem 4.3 can be extended to the situation where the number of objective function observations collected at the current or candidate solution (K) can depend deterministically on the current and candidate solutions.

4.4.2 Simulated Annealing with Averaging

In this section we present our second variant of the SA algorithm and also state and prove its convergence properties. This method employs averaging. Consequently, it is non-Markovian and time-inhomogeneous and, to the best of our knowledge, it is the first such a variant of the SA algorithm that is rigorously proved to be convergent under assumptions that are verifiable in practice. Also, this approach is adaptive to the information gathered so far, a fact that can have a considerable impact on the empirical performance (see Section 4.5

below). Given that this method is within our framework presented in Section 4.2.2, this section also provides an example of application of Theorem 4.2. Finally, we show that the sequence of current iterates generated by this variant also converges in probability to the set of globally optimal solutions. We are now ready to state our algorithm:

Algorithm 4.4

Step 0: Identical to Step 0 of Algorithm 4.2.

Step 1: Identical to Step 1 of Algorithm 4.2.

Step 2: Given $\theta_n = \eta$ and $\theta'_n = \eta'$, generate additional observations $\{X_\eta^i\}_{i=C_n(\eta)+1}^{C_n(\eta)+K_n(\eta)}$ of X_η and $\{X_{\eta'}^i\}_{i=C_n(\eta')+1}^{C_n(\eta')+K_n(\eta')}$ of $X_{\eta'}$. For $\theta = \eta, \eta'$, let

$$\Sigma_{n+1}(\theta) = \Sigma_n(\theta) + \sum_{i=C_n(\theta)+1}^{C_n(\theta)+K_n(\theta)} h_\theta(X_\theta^i)$$

and $C_{n+1}(\theta) = C_n(\theta) + K_n(\theta)$. Moreover, let $C_{n+1}(\theta) = C_n(\theta)$ and $\Sigma_{n+1}(\theta) = \Sigma_n(\theta)$ for all $\theta \in \Theta$, $\theta \neq \eta, \eta'$. Calculate $\hat{f}_{n+1}(\theta) = \Sigma_{n+1}(\theta)/C_{n+1}(\theta)$ for $\theta = \eta, \eta'$.

Step 3: Identical to Step 3 of Algorithm 4.3.

Step 4: Identical to Step 4 of Algorithm 4.2.

The main difference between Algorithms 4.3 and 4.4 is that the estimate of the objective function value $\hat{f}_{n+1}(\theta)$ at the candidate or current solution θ in iteration n that is used to decide on the next current solution is the average of all observations collected at θ so far in Algorithm 4.4, as opposed to only the average of observations collected in the current iteration in Algorithm 4.3. This modification allows us to weaken the assumption on the estimated objective function values (i.e., it now suffices that they be strongly consistent, rather than averages of independent, identically distributed, and unbiased observations) and still maintain the convergence guarantee of Algorithm 4.3 (under the additional conditions given in Assumptions 4.3 and 4.4). Also, if θ is a candidate or current solution, then the number of observations collected at θ in iteration n is $K_n(\theta)$, which can be chosen adaptively as long as it satisfies the following assumption.

Assumption 4.15. *The number of objective function observations at the current or candidate solutions in iteration n, $K_n(\theta_n)$ and $K_n(\theta'_n)$, depend only on the information gathered by the algorithm in the first $n-1$ iterations. Moreover, $K_n(\theta) > 0$ when $C_n(\theta) = 0$ and $\theta \in \{\theta_n, \theta'_n\}$.*

Algorithm 4.4 differs from the SA algorithm of Fox and Heine [31] in the choice of estimator of the optimal solution, and hence in the mode of convergence. The convergence analysis presented by Fox and Heine [31] shows that the sequence of current solutions $\{\theta_n\}$ generated by their variant of the SA algorithm converges in probability to the set Θ^* provided that each feasible solution is sampled infinitely often with probability one and the SA algorithm for deterministic optimization converges in probability to the set Θ^*. Rather than assuming that each solution is sampled infinitely often with probability one, we provide conditions under which this happens. In fact, this is one of the major contributions of this chapter.

Let

$$L = \max_{\theta \in \Theta} \max_{\theta' \in N(\theta)} [f(\theta) - f(\theta')]^+. \tag{4.7}$$

Next we prove our main convergence result for Algorithm 4.4.

Theorem 4.4. *Suppose that Assumptions 4.1 through 4.5, 4.11, 4.12, and 4.15 are satisfied and that the cooling schedule $\{T_n\}$ satisfies equation (4.6), where L is defined in equation (4.7). Then the sequence $\{\theta^*_n\}$ generated by Algorithm 4.4 converges almost surely to the set Θ^* in the sense that for almost every $\omega \in \Omega$, there exists $N(\omega)$ such that $n \geq N(\omega)$ implies that $\theta^*_n(\omega) \in \Theta^*$.*

Proof: Note that Algorithm 4.4 is a special case of Algorithm 4.2. By Theorem 4.2 it suffices to verify that Assumptions 4.9 and 4.10 are satisfied by Algorithm 4.4. From the statement of Algorithm 4.4, it is clear that Assumption 4.9 holds. Also, observe that without loss of generality we can assume that Assumption 4.14 holds. But then Proposition 4.1 applied to deterministic optimization implies that Assumption 4.10 holds (note that Assumption 4.13 holds and equations (4.3) and (4.7) coincide for deterministic optimization). ■

Note that by Jensen's inequality, we have that L defined in equation (4.3) is larger than L defined in equation (4.7). Hence Algorithm 4.4 is convergent with lower C values (see Remark 4.5) than Algorithm 4.3 under Assumption 4.13. The next corollary shows that the sequence of current iterates $\{\theta_n\}$ generated by Algorithm 4.4 converges in probability to Θ^*.

Corollary 4.2. Suppose that Algorithm 4.4 for optimizing a deterministic objective function f converges in probability to Θ^* (for conditions under which this occurs see Hajek [42] and Tsitsiklis [88]) and the conditions in Theorem 4.4 are satisfied. Then the sequence $\{\theta_n\}$ generated by Algorithm 4.4 converges in probability to Θ^*.

Proof: Without loss of generality, we can assume that Assumption 4.14 holds. Let X be a Markov chain of current solutions generated by Algorithm 4.4 for deterministic optimization and $Y = \{\theta_n\}$. For each $k \in \mathbb{N}$, let $Z(k, \cdot) = \{Z(k, n)\}$ be a discrete time stochastic process that coincides with Y up to time k for every $\omega \in \Omega$ (i.e., $Z(k, n, \omega) = \theta_n(\omega)$ for all $n \leq k$ and $\omega \in \Omega$) and behaves probabilistically the same as the Markov chain X for $n \geq k$. Let N be as in the proof of Theorem 4.1 (i.e., an iteration number such that if $n \geq N$, then $\hat{f}_n(\theta) = f(\theta)$ for all $\theta \in \Theta$). From the proofs of Theorems 4.2 and 4.4, it is clear that N is almost surely finite. Also, we may assume that $Z(k, n, \omega) = \theta_n(\omega)$ for all $n \in \mathbb{N}$ provided that $N(\omega) \leq k$. Observe that this setup satisfies conditions (i) through (iii) in Fox and Heine [31]. Careful analysis of the proof of Theorem 1 in Fox and Heine [31] shows that the conclusion of this theorem still holds with our setup (the original analysis assumes that Y and $\{Z(n, \cdot) : n \in \mathbb{N}\}$ are Markov chains, while in our situation it is obvious that they are not Markov chains in general). This shows that $\{\theta_n\}$ generated by Algorithm 4.4 converges in probability to Θ^*. ■

4.4.3 Simulated Annealing with Averaging and Uncountable Precision

In this section we discuss how Assumption 4.3(which can be viewed as a *countable precision* assumption in the knowledge of the objective function estimates) can be relaxed for the SA algorithm with averaging (Algorithm 4.4). We will need the following assumption.

Assumption 4.16. *The number of objective function observations at the candidate and*

current solutions, $K_n(\theta_n)$ and $K_n(\theta'_n)$, depend only on all the objective function observations collected at the candidate and current solutions by the algorithm in the first $n-1$ iterations, the number of such observations $C_n(\theta_n)$ and $C_n(\theta'_n)$, and the iteration number n. Moreover, $K_n(\theta) > 0$ when $C_n(\theta) = 0$ and $\theta \in \{\theta_n, \theta'_n\}$.

Observe that Assumption 4.16 states that $K_n(\theta_n)$ and $K_n(\theta'_n)$ can now depend only on "local" information as opposed to the information about all feasible points (see Assumption 4.15). We will also require that the cooling schedule satisfies the following condition

$$\sum_{n=0}^{\infty} \exp\left(-\frac{rL'}{T_n}\right) = +\infty, \tag{4.8}$$

where $L' > L$ with L is defined in (4.7). We now present our convergence result.

Theorem 4.5. *Suppose that Assumptions 4.1, 4.2, 4.4, 4.5, 4.11, 4.12, and 4.16 are satisfied and the cooling schedule satisfies equation (4.8). Then the sequence $\{\theta^*_n\}$ generated by Algorithm 4.4 converges almost surely to the set Θ^* in the sense that for almost every $\omega \in \Omega$, there exists $N(\omega)$ such that $n \geq N(\omega)$ implies that $\theta^*_n(\omega) \in \Theta^*$.*

We now briefly outline the proof of Theorem 4.5. Let $\tilde{\Omega}_s$ and $\bar{\Omega}$ be as defined in the proof of Theorem 4.1. First, for each $\omega_s \in \tilde{\Omega}_s$, we identify a probability one subset $\tilde{\Omega}_d(\omega_s) \subset \Omega_d$ under which the SA algorithm with averaging visits each feasible solution infinitely often, provided that it is initialized with a "sufficient" number of observations collected at each point and these observations are collected under ω_s. The proof of the fact that $\tilde{\Omega}_d(\omega_s)$ is a probability one subset relies on extending Lemma 4.2 and Proposition 4.1. Then we show that the SA algorithm with averaging converges to the set Θ^* under $\omega \in (\tilde{\Omega}_d(\omega_s) \times \{\omega_s\}) \cap \bar{\Omega}$. Finally, we show that $\mathbb{P}(\tilde{\Omega}) = 1$, where $\tilde{\Omega} = \cup_{\omega_s \in \tilde{\Omega}_s} \tilde{\Omega}_d(\omega_s) \times \{\omega_s\}$. The details of the proof of Theorem 4.5, together with a discussion of how Assumption 4.16 can be weakened, are given in Appendix B.

The main reason why we specify ω_s and then identify $\tilde{\Omega}_d(\omega_s)$ in this case, as opposed to directly specifying $\tilde{\Omega}_d$ that works for every $\omega_s \in \tilde{\Omega}_s$ as we did in Theorems 4.1 and 4.2, is that in the presence of uncountable precision, the exact coupling of Algorithm 4.4 for stochastic optimization with Algorithm 4.4 for deterministic optimization is not possible because the

objective function value estimates at each feasible point do not converge almost surely in a finite number of iterations. Instead, we achieve exact coupling for each $\omega \in (\tilde{\Omega}_d(\omega_s) \times \{\omega_s\}) \cap \bar{\Omega}$. To the best of our knowledge, this approach to proving the convergence of a random search method for simulation optimization is novel. Our approach is particularly interesting because by fixing the randomness arising from the simulation of objective function values (ω_s), we have converted the convergence analysis of a non-Markovian algorithm to the study of a collection of Markov chains. Such an approach can be useful in analyzing non-Markovian algorithms because exploiting the Markov chain structure usually facilitates proving the convergence of the method directly.

We conclude this section by comparing SA algorithms with averaging implemented under the assumptions of Theorems 4.4 and 4.5. We do not consider the restriction that the cooling schedule satisfy (4.8) rather than (4.6) to be crucial because it is natural to estimate the quantity L via an upper bound. On the other hand, Assumption 4.16 is more restrictive because it does not allow the method to use all the information gathered so far. This restriction is an important part of the convergence proof (because it facilitates coupling). Although these restrictions provide a mathematically more elegant set of assumptions under which the SA algorithm with averaging is guaranteed to converge, we think that in practice it is not important to ensure that Assumption 4.16 holds, as long as Assumption 4.15 is satisfied. The reason is that we think Assumption 4.3 is not restrictive from a practical point of view (as we have discussed before).

4.5 Numerical Examples

In this section, we present two numerical examples that illustrate the performance of Algorithms 4.3 and 4.4. These examples show that the adaptiveness of the methods with averaging can be quite useful from a practical point of view. In Section 4.5.1, we give an example of a deterministic problem with white noise added and in Section 4.5.2, we provide an example from a manufacturing context.

4.5.1 Two Hills Problem

In this section, we present numerical results obtained by applying Algorithms 4.3 and 4.4 to solve the *two hills* problem described in Section 3.5.1 with $\sigma^2 = 50$. Notice that the standard deviation of the white noise is roughly equal to the range of the objective function values. This makes the response surface highly noisy and hence this problem is relatively difficult to solve.

We next describe the implementation details. We first define two different neighborhood structures. For each $\theta \in \Theta$, let

$$N_L(\theta) = \{(y_1, y_2) \in \Theta \setminus \{\theta\} : |y_i - x_i| \leq 1 \text{ for } i = 1, 2\}$$

and $N_G(\theta) = \Theta \setminus \{\theta\}$. For all $n \in \mathbb{N}$ and $\theta \in \Theta$, $Q_n(\theta, \cdot)$ in iteration n is the uniform distribution on $N_L(\theta)$ and $N_G(\theta)$ for the Local and Global SA algorithms, respectively. The cooling schedule $\{T_n\}$ for each optimization method in our experiments is of the form $T_n = C/\log(n + 10)$ for all $n \in \mathbb{N}$, where C is a positive constant. The parameters in our experiments are chosen to guarantee the almost sure convergence of each method (see Theorems 4.3 and 4.4), and are given in Table 4.1. Note that the cooling schedules selected for the Global and Local SA algorithms are the same across Algorithms 4.3, 4.4, and 4.4A, and, hence, the difference in the performances of the methods can be attributed to the ways in which they estimate the objective function values at the current and candidate solutions in order to decide on the next current point (i.e., no averaging versus averaging and also adaptive versus non-adaptive number of observations collected at the current and candidate solutions).

We next comment on the choice of parameters in Table 4.1. The C values are estimated via reasonably tight upper bounds on the product rL for Local and Global Algorithms 4.3, and the same C values are also used for Local and Global Algorithms 4.4 and 4.4A. In particular, r is bounded by twice the diameter of the graph G and L in (4.3) is bounded as

Table 4.1: Parameters for each method on the two hills problem

Algorithm	C	K	K_n
Local Algorithm 4.3	565	10	
Global Algorithm 4.3	20	10	
Local Algorithm 4.4	565		10
Global Algorithm 4.4	20		10
Local Algorithm 4.4A	565		Adaptive
Global Algorithm 4.4A	20		Adaptive

follows:

$$\begin{aligned} L &= \max_{\theta\in\Theta}\max_{\theta'\in N(\theta)} \mathbb{E}[N(f(\theta)-f(\theta'),2\sigma^2/K)]^+ \\ &\leq \max_{\theta\in\Theta}\max_{\theta'\in N(\theta)} |f(\theta)-f(\theta')| + \mathbb{E}[|N(0,2\sigma^2/K)|] \qquad (4.9) \\ &= \max_{\theta\in\Theta}\max_{\theta'\in N(\theta)} |f(\theta)-f(\theta')| + \sqrt{\frac{4\sigma^2}{\pi K}}. \end{aligned}$$

Note that the above derivation, together with Jensen's inequality, provides guidance on the magnitude of L in practice (when f and the variances of $h_\theta(X_\theta)$, where $\theta \in \Theta$, are unknown). In particular, if the first term in (4.9) is bounded by F_N and $\sup_{\theta\in\Theta} Var\{h_\theta(X_\theta)\} \leq \sigma^2$, then L in (4.3) is bounded by $F_N + \sqrt{2\sigma^2/K}$.

We choose relatively large values for K for Algorithm 4.3 and $K_n(\theta)$ for Algorithm 4.4, where $n \in \mathbb{N}$ and $\theta \in \Theta$, because of the high variance of X_θ. Finally, let $\hat{\sigma}_n^2(\theta)$ be a standard variance estimator of a single objective function observation at a solution θ obtained using all the data collected on $f(\theta)$ in the first n iterations. Then the procedure for selecting $K_n(\theta)$ for $\theta \in \{\theta_n, \theta_n'\}$ in Local and Global Algorithms 4.4A is given in Algorithm 4.5.

We next briefly explain how Algorithm 4.5 works. We let $K_n(\theta) = k_0 > 0$ provided that no more than one objective function observation has been collected at θ before. Otherwise, with probability α (usually small), $K_n(\theta) = k_1 > 0$. With the remaining probability of $1-\alpha$, we statistically estimate how likely it is that $f(\theta)$ is better than $f(\theta_n^*)$ (this is controlled by the parameter β). If it is sufficiently likely to be the case, then we calculate if it would be more desirable to invest k_1 observations into θ or θ_n^* in order to reduce the variance of the estimator of the difference in the objective function values at θ and θ_n^*. Note that

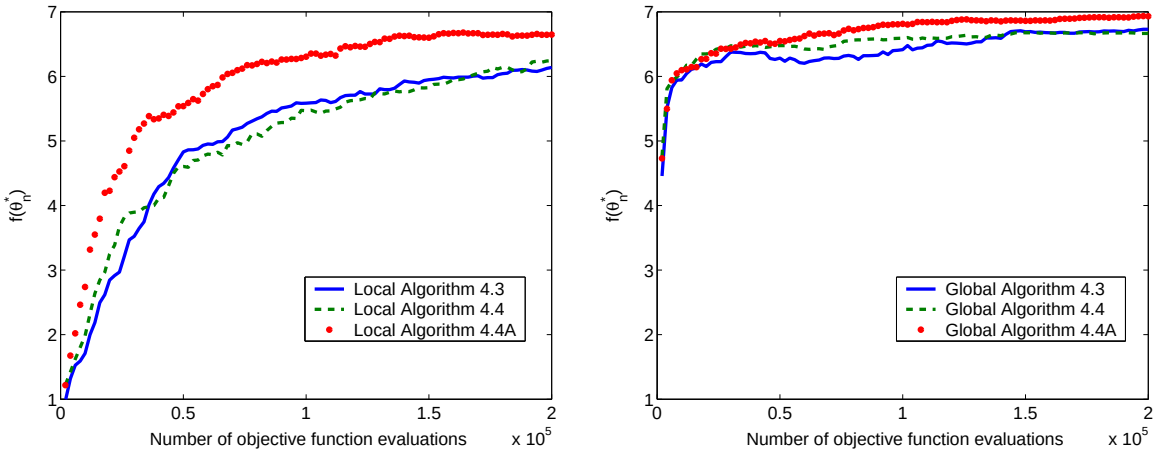
Algorithm 4.5 Adaptive Selection of the Number of Observations

1: $K_n(\theta) = 0$
2: **if** $C_n(\theta) \leq 1$ **then**
3: $\quad K_n(\theta) = k_0$
4: **else if** $Uniform(0,1) \leq \alpha$ **then**
5: $\quad K_n(\theta) = k_1$
6: **else**
7: $\quad z = \{\hat{f}_n(\theta_n^*) - \hat{f}_n(\theta)\}/\{\hat{\sigma}_n^2(\theta_n^*)/C_n(\theta_n^*) + \hat{\sigma}_n^2(\theta)/C_n(\theta)\}$
8: $\quad$ **if** $z \leq \beta$ **then**
9: $\quad\quad \sigma_1 = \hat{\sigma}_n^2(\theta_n^*)/(C_n(\theta_n^*) + k_1) + \hat{\sigma}_n^2(\theta)/C_n(\theta)$
10: $\quad\quad \sigma_2 = \hat{\sigma}_n^2(\theta_n^*)/C_n(\theta_n^*) + \hat{\sigma}_n^2(\theta)/(C_n(\theta) + k_1)$
11: $\quad\quad$ **if** $\sigma_1 \geq \sigma_2$ **then**
12: $\quad\quad\quad K_n(\theta) = k_1$
13: $\quad\quad$ **end if**
14: $\quad$ **end if**
15: **end if**

investing into θ might not be desirable if the variance of $\hat{f}_n(\theta)$ is already very low (i.e., more observations will not provide much more information about $f(\theta)$). In this case, we let $K_n(\theta) = 0$. On the other hand, if it is desirable to invest into θ, we let $K_n(\theta) = k_1$. Observe that even though it might be more desirable to invest k_1 observations into θ_n^*, we do not do so because in SA, observations are collected only at the current and candidate solutions, and θ_n^* might not be one of them. For this numerical experiment, we use Algorithm 4.5 with $k_0 = k_1 = 10$, $\alpha = 0.2$, and $\beta = 2.33$ (i.e., the 99 percent quantile of a standard normal random variable). Observe that Assumption 4.5 holds when the $\{K_n(\theta)\}$ are chosen in this way.

The initial solution is selected randomly for all six algorithms. The performance of the algorithms is compared based on 100 independent replications. We used common random numbers in our experiment in the sense that the initial seeds for the sequences of uniforms required for choosing θ_0 in Step 0 of the algorithms, generating θ_n' in Step 1, estimating the objective function values in Step 2, and selecting θ_{n+1} in Step 3 are the same. Figure 4.1 shows the average performance of the six approaches as the simulation effort increases.

It is clear from part (a) of Figure 4.1 that Local Algorithm 4.4A performs better than Local Algorithms 4.3 and 4.4. Also, from part (b) of Figure 4.1, it is obvious that Global Algorithm 4.4A dominates Global Algorithm 4.3 and is significantly better than Global



(a) Local Algorithms 4.3, 4.4, and 4.4A (b) Global Algorithms 4.3, 4.4, and 4.4A

Figure 4.1: Performance of the local and global algorithms on the two hills problem

Algorithm 4.4. This shows that the flexibility provided by averaging (the fact that the number of objective function observations at the current and candidate solutions can depend on the information seen by the algorithm so far) can be quite useful in practice. In fact, for these numerical studies, we have only tried two different adaptive strategies for selecting $K_n(\theta)$ and hence it is likely that the strategy provided in Algorithm 4.5 is far from the "best" one. The identification of improved strategies for choosing the parameters $K_n(\theta)$, however, is beyond the scope of this work. Observe that Local and Global Algorithms 4.4A do not satisfy Assumption 4.16 of Theorem 4.5 but they do satisfy the assumptions of Theorem 4.4. These and the subsequent numerical results suggest that implementing Algorithm 4.4 under the assumptions of Theorem 4.4, rather than those of Theorem 4.5, may be beneficial.

It is easy to see from part (a) of Figure 4.1 that Local Algorithm 4.4 initially performs better, then worse, and then again better than Local Algorithm 4.3. There are two reasons for such a behavior, namely that this problem has locally optimal solutions and that the estimates of objective function values at the current and candidate solutions are less noisy for Local Algorithm 4.4 than for Local Algorithm 4.3. Thus, it is likely that Local Algorithm 4.4 identifies a local solution faster, yielding better behavior in the beginning of the search. But because Algorithm 4.4 uses more precise objective function estimates, it is more likely to

get stuck at nonoptimal solution, yielding worse behavior afterwards. Moreover, once Local Algorithm 4.4 escapes this local solution, it is likely to identify a global optimal solution faster than Local Algorithm 4.3, and, therefore, yield better performance toward the end of the search. From part (b) of Figure 4.1, it is clear that Global Algorithm 4.4 performs better than Global Algorithm 4.3, and the reason for this is that due to the smaller noise in $\hat{f}_n(\theta_n)$ and $\hat{f}_n(\theta'_n)$, it is easier for Global Algorithm 4.4 to identify optimal or nearly optimal solutions.

Observe that higher noise in the estimated objective function values has similar effects as higher temperature values in that both imply that the SA algorithm will be moving more aggressively within the entire feasible region. In other words, there is a correspondence between the noise in the objective function estimates and the value of the temperature parameter. Because in general it is not obvious whether a particular value for the temperature is better at a given stage of a search, it is also not possible to assert in advance whether averaging alone is going to be beneficial relative to no averaging. This conclusion is supported by the numerical results in this and the next section.

It is also obvious from Figure 4.1 that the global versions of our algorithms perform considerably better in this example than their local counterparts. This can be explained by the fact that an initial solution might be far away from the subregions containing good solutions and it might take the local algorithms many iterations to identify a good subregion.

We also tried all six approaches on this problem using different parameter settings. We found that the multiplier C in the cooling schedule does not have a significant impact on the performance of all six approaches (as long as it is in some reasonable range). On the other hand, the number of observations collected at the current and candidate solutions (we tried $K = K_n(\theta) = k_0 = k_1 = 2$ in Table 4.1, rather than $K = K_n(\theta) = k_0 = k_1 = 10$) has a larger impact on the performance. Despite this, the relative performance of the methods is unaffected, and hence the numerical results in this section are representative for our SA algorithms on the two-hills problem.

4.5.2 Three-Stage Buffer Allocation Problem

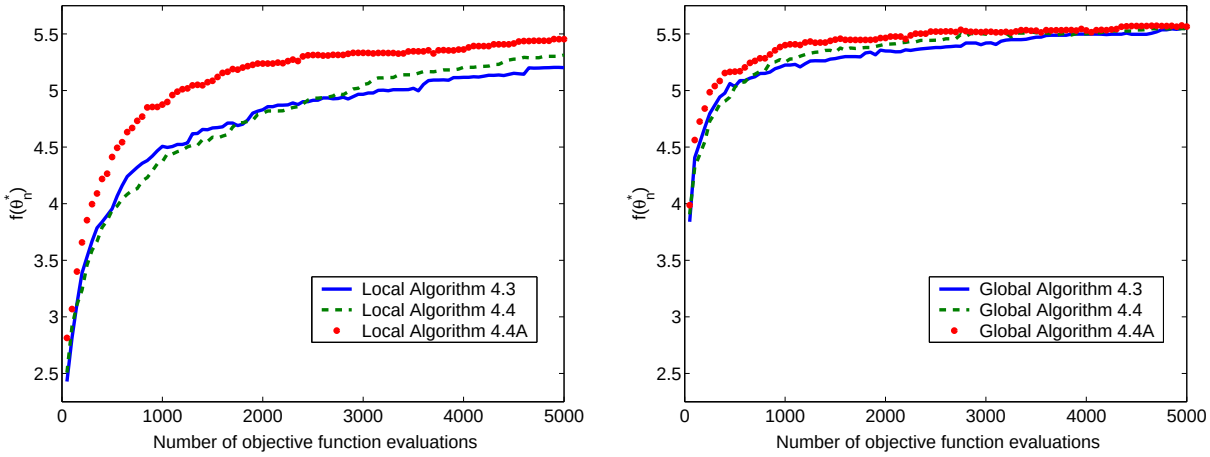
In this section, we present numerical results for Algorithms 4.3 and 4.4 when applied to solve the stochastic version of the *three-stage buffer allocation* problem defined in Section 3.5.1. We start by discussing the implementation details. We define a local neighborhood $N_L(\theta)$ of a feasible solution θ as the set of feasible points that can be obtained by shifting a single buffer slot between buffers, increasing or decreasing service rate by 1 at a single workstation, or shifting a single unit of service rate between two workstations. We let the global neighborhood $N_G(\theta)$ be defined as before. As in Section 4.5.1, for all $n \in \mathbb{N}$ and $\theta \in \Theta$, $Q_n(\theta, \cdot)$ in iteration n is the uniform distribution on $N_L(\theta)$ and $N_G(\theta)$ for the Local and Global SA algorithms, respectively. The cooling schedule $\{T_n\}$ is again of the form $T_n = C/\log(n + 10)$ for all $n \in \mathbb{N}$, where C is a positive constant. As in Section 4.5.1, the parameters in our experiments are chosen based on Theorems 4.3 and 4.4 to guarantee the almost sure convergence of each method, and are given in Table 4.2. As in Section 4.5.1, note that the cooling schedule $\{T_n\}$ is the same for all three local (global) methods to facilitate comparison.

Table 4.2: Parameters for each method on the three-stage buffer allocation problem

Algorithm	C	K	K_n
Local Algorithm 4.3	280	3	
Global Algorithm 4.3	14	3	
Local Algorithm 4.4	280		3
Global Algorithm 4.4	14		3
Local Algorithm 4.4A	280		Adaptive
Global Algorithm 4.4A	14		Adaptive

We now comment on the choice of the parameters. The C values are again estimated via reasonably tight upper bounds on the product rL (r is again taken to be twice the diameter of the graph G, while L is taken to be 1.5 times the value of L that is calculated based on equation (4.7)). We choose relatively small values for K for Algorithm 4.3 and $K_n(\theta_n)$ and $K_n(\theta'_n)$ for Algorithm 4.4, where $n \in \mathbb{N}$ and $\theta \in \Theta$, because of the low variance of $h_\theta(X_\theta)$. The number of objective function observations collected at the current and

candidate solutions in iteration n for Algorithms 4.4A is chosen based on Algorithm 4.5 with $k_0 = k_1 = 3$, $\alpha = 0.2$, and $\beta = 2.33$. Note again that Assumption 4.5 is satisfied when the sequences $\{K_n(\theta)\}$ are chosen in this way. The initial solution is selected randomly for all six methods and the performance of the algorithms is compared based on 50 independent replications. Common random numbers are employed similarly as in Section 4.5.1. Figure 4.2 shows the average performance of the six approaches as the simulation effort increases.



(a) Local Algorithms 4.3, 4.4, and 4.4A (b) Global Algorithms 4.3, 4.4, and 4.4A

Figure 4.2: Performance of the local and global algorithms on the three-stage buffer allocation problem

It is clear from parts (a) and (b) of Figure 4.2 that Global and Local Algorithms 4.4A perform better than their counterparts of both Algorithm 4.3 and 4.4. This again demonstrates that averaging together with choosing the number of observations collected at the current and candidate solutions adaptively can be efficient numerically. From the performance of Algorithms 4.3 and 4.4 (see parts (a) and (b) of Figure 4.2), we again see that averaging alone is not necessarily beneficial on this problem (see the discussion in Section 4.5.1).

Also notice that the Global Algorithms perform considerably better in this example than their local counterparts. The reason is that the temperature values are higher for the Local Algorithms, and, hence, the sequence of current iterates generated by the local methods tends to move around within the feasible region rather than settling down in a region with

good alternatives (and thus, the local methods fail to obtain more precise objective function estimates at good points). We also tested all six methods on this problem using a different multiplier C and different $K = K_n(\theta) = k_0 = k_1$. As in Section 4.5.1, we found that the performance of all approaches is not significantly impacted by the value of C, with the value of $K = K_n(\theta) = k_0 = k_1$ having a more substantial impact on performance.

4.6 Conclusions

In this chapter, we presented a general framework based on averaging for designing adaptive and almost surely convergent random search methods for discrete simulation optimization. The objective function estimate at any solution is the average of all observations collected at this solution so far. The methods are adaptive in the sense that they use all information obtained so far by the search algorithm to decide on how to expend simulation effort on the sampled points. We also presented two new variants of the SA algorithm and discussed their convergence. These analyses provided increased theoretical understanding of SA with decreasing cooling schedule for deterministic and stochastic optimization. Moreover, via numerical examples involving the proposed SA algorithms, we demonstrated that averaging together with adaptiveness in expending simulation effort can be effective.

CHAPTER V

ADAPTIVE RANDOM SEARCH FOR CONTINUOUS STOCHASTIC OPTIMIZATION

5.1 Introduction

In this chapter we consider continuous simulation optimization problems. Most of the existing research aimed at solving such problems involves estimating the gradient (and possibly higher order derivatives) of the objective function f. This includes methods like stochastic approximation and SAA (see Chapter 2 for a short review of these methods).

In this chapter we adopt a different approach to solving the problem (1.1) in that we do not use gradient information. We do this for a number of reasons, including the fact that the objective functions of some continuous simulation optimization problems may not have gradients, or these gradients may be difficult or expensive to estimate, rendering methods like stochastic approximation and SAA difficult to apply. Instead our approach is based on random search. Unfortunately, most of the existing work on random search for simulation optimization is done for discrete settings, where Θ is a finite element set. The convergence of such methods is usually ensured by showing that promising solutions (including the optimal solution) are sampled repeatedly (so that the noise in the objective function estimates is eventually reduced). This property is difficult to achieve in the continuous simulation optimization setting, and hence special techniques are required for solving such problems. The three approaches considered in this chapter reduce the effects of noise either by occasional resampling of already sampled solutions or by averaging observations in balls that shrink with time.

More specifically, in this part of the thesis, we present and analyze three random search methods for solving continuous simulation optimization problems. The main difference between the methods involves the way they estimate objective function values, and hence the approach they use to control noise. Moreover, one of the approaches is adaptive, while

the other two are based on pure random search. We also present conditions under which the three methods are convergent, both in probability and almost surely. Finally, we numerically demonstrate the effectiveness of the three methods when compared to some other random search methods.

We now describe our adaptive search with resampling (ASR) approach. In certain iterations, this method adaptively samples new solutions in Θ, then, based on an acceptance criterion, decides (with the goal of retaining only promising points) whether the newly sampled point should be included in the set of accepted sampled points, and finally ensures that each accepted sampled point has "enough" observations collected at it. At other times, the method adaptively resamples solutions from the set of accepted sampled points with the goal of comparing the quality of these points, and hence improving the estimator of the optimal solution. This method is the main contribution of this work because it is not only adaptive (in that some algorithmic decisions may be based on all the information collected by the method so far) and provably convergent, but also exhibits good empirical performance.

We also study a deterministic shrinking ball (DSB) algorithm. This method is based on pure random search and was first proposed and analyzed by Baumert and Smith [20]. The estimate of the objective function value at each sampled solution θ in iteration k is the average of all objective function observations collected at sampled points that are at most a distance r_k away from θ, with r_k decreasing to zero as k grows. Our contribution lies in the generality of our convergence analysis and in being the first to document the numerical performance of the method. Consequently, our work provides an increased theoretical and practical understanding of the method.

Finally, we propose a stochastic shrinking ball (SSB) algorithm that resembles the DSB algorithm, with the only difference being that the estimate of the objective function value at each sampled solution θ in iteration k is the average of the objective function observations at the n_k sampled points that are closest to θ. Although the numerical performance of the DSB and SSB algorithms is usually similar (see Section 5.5.3), our experience is that when there is a noticeable difference in performance, the SSB method outperforms the DSB

method. Also, in practice it may be easier to choose the sequence $\{n_k\}$ for the SSB method rather than the sequence $\{r_k\}$ for the DSB method.

The remainder of this chapter is organized as follows. In Section 5.2, we present our ASR method and discuss its convergence. In Section 5.3, we present the DSB algorithm and provide its convergence analysis, while in Section 5.4, we present our SSB algorithm and discuss conditions under which it converges. In Section 5.5, we provide some numerical results that demonstrate the effectiveness of our approaches when compared to other random search methods available in the literature. Finally, concluding remarks are given in Section 5.6.

5.2 Adaptive Search with Resampling

In this section we present and analyze our first random search method for continuous optimization. More specifically, in Section 5.2.1 we present our ASR method, and in Section 5.2.2 we give its convergence analysis. Finally, in Section 5.2.3 we provide discussion on how assumptions under which our method is guaranteed to converge can be satisfied in practice.

5.2.1 Algorithm Description

In this section we present our first algorithm. We start by introducing some notation. For all $\theta \in \Theta$ and $k \in \mathbb{N}$, let $N_k(\theta)$ be the number of objective function observations collected at θ by the end of iteration k and let $S_k(\theta)$ be the sum of these $N_k(\theta)$ objective function observations. Also, for all $\theta \in \Theta$ and $k \in \mathbb{N}$, let $\hat{f}_k(\theta) = S_k(\theta)/N_k(\theta)$. Let Θ_k be the set of solutions sampled and accepted by the end of iteration k. Finally, let $\{K(i)\}$ be a nondecreasing sequence of positive integers, and let $M(i) = \lfloor i^b \rfloor$ for $i = 1, 2, \ldots$, where $b \geq 1$. The pseudo-code for our ASR method is given in Algorithm 5.1.

We now briefly comment on our algorithm. At each iteration of the method, one of two sets of actions takes place. The first set of actions occurs if the current iteration number is equal to some element in the sequence $\{M(i)\}$. In this case, we sample a new solution θ from Θ using the sampling procedure. This step is intended for adaptively searching the entire feasible region for improved solutions. Then, based on some acceptance criterion, we decide whether we want to include the sampled point in the set of sampled and accepted

Algorithm 5.1 Adaptive Search with Resampling (ASR) Algorithm

1: Select $b \geq 1$, a sampling strategy, a resampling strategy, an acceptance criterion, and a sequence $\{K(i)\}$. Let $i = 1$ and $k = 0$.
2: **while** Stopping criterion is not satisfied **do**
3: Let $k = k + 1$
4: **if** $k = M(i)$ **then**
5: Sample a solution θ_i from Θ using the sampling strategy
6: Based on the acceptance criterion, decide whether to include θ_i into the set of sampled points by iteration k, Θ_k, so that $\Theta_k \in \{\Theta_{k-1}, \Theta_{k-1} \cup \{\theta_i\}\}$ and update $N_k(\theta_i)$ and $S_k(\theta_i)$ if needed (any observations of $f(\theta_i)$ are collected independent of everything else)
7: For each $\theta \in \Theta_k$, if $N_k(\theta) < K(i)$, obtain $K(i) - N_k(\theta)$ additional independent objective function observations of $f(\theta)$ and update $N_k(\theta)$ and $S_k(\theta)$ accordingly
8: Let $i = i + 1$
9: **else**
10: Sample a solution θ from Θ_{k-1} using the resampling strategy
11: Obtain an estimate of the objective function value at θ independent of everything else and update $N_k(\theta)$ and $S_k(\theta)$
12: **end if**
13: **end while**
14: Select an estimate of the optimal solution $\theta_k^* \in \arg\max_{\theta \in \Theta_k} \hat{f}_k(\theta)$

points. The idea is to include points that appear to have high objective function values and reject others. This is important because for the convergence of the method, we require the number of objective function observations collected at each accepted sampled point to grow at least at the rate of $K(i)$, where i is the number of sampled points, and, thus, discarding "bad" points can save a considerable amount of simulation effort. Finally, we ensure that each accepted sampled point has a sufficient number (i.e., at least $K(i)$) of objective function observations collected at it.

The second set of actions is comprised of sampling a solution from the set of sampled and accepted points using the resampling strategy and obtaining an objective function observation at that point. The goal is to allocate simulation effort to points that look attractive. By doing so, we intend to improve the estimator θ_k^* of the optimal solution, and hence improve the empirical performance of our method. The reason why it is better to allocate simulation effort to improve the objective function estimates at "good" points, rather than "bad" points, is that it is easier to differentiate between "good" and "bad" points than between two "good" points. For a more thorough discussion of this topic, the

interested reader is referred to Section 3.2 .

When compared to other methods in the literature, our ASR method resembles the method of Yakowitz and Lugosi [94] (YL) to the greatest extent. On the algorithmic side, the differences are that our sampling strategy can be adaptive (and hence may include a local search component), our method incorporates an acceptance criterion, we use a different estimator of the optimal solution, and our resampling strategy can be more aggressive. In the YL method, new solutions are always sampled from a specified distribution (as in a global search that does not adapt to the information obtained), and hence it might take a long time to identify good solutions. Moreover, every sampled solution is always accepted by the YL method, and hence it can be quite wasteful in the use of the available simulation budget. Also, their estimator of the optimal solution is the most recently sampled point (using either the sampling strategy or the resampling strategy). In contrast, our estimator of the optimal solution is the accepted point that has the highest estimated objective function value. Also, because we use a different estimator of the optimal solution, our resampling strategy can be more aggressive (it can be specified arbitrarily by a user), while they require their resampling procedure (distribution) to be closely related to a Boltzmann-type distribution that converges to the set of global optimal solutions as the number of iterations grows. On the theoretical side, we show that our method is convergent both in probability and almost surely, while they claim convergence in probability only. Indeed, it easily can be seen that their method cannot converge almost surely because the sampling procedure is used infinitely often and the most recently sampled solution is the estimator of the optimal solution. Finally, in Section 5.5 we demonstrate that ASR behaves better empirically than the YL method.

5.2.2 Convergence Analysis

Before presenting our main convergence result for the ASR algorithm, we provide the following lemma. Although similar results exist in the literature, with the special case of $l = 2$ proved by Baumert and Smith [20], we have not found a result that implies ours, and hence we provide the following lemma, together with its proof, for completeness.

Lemma 5.1. *Let $\{Z_i\}_{i=1}^{\infty}$ be a sequence of independent random variables with mean zero such that $\mathbb{E}[Z_i^{2l}] \leq R < \infty$ for $i = 1, 2, \ldots$ and $l \in \mathbb{N}^+$. Let $S_n = \sum_{i=1}^{n} Z_i$. Then for each $\epsilon > 0$ there exists $C \in \mathbb{R}$ such that $\mathbb{P}(|S_n| \geq \epsilon n) \leq C/n^l$ for all $n \in \mathbb{N}^+$.*

Proof: Fix $\epsilon > 0$ and $n \geq l$. By Markov's inequality we have that

$$\mathbb{P}(|S_n| \geq \epsilon n) \leq \frac{\mathbb{E}[S_n^{2l}]}{(\epsilon n)^{2l}}. \tag{5.1}$$

Moreover, fix $k \leq 2l$. Note that $|x|^{2l/k}$ is a convex function. Thus, Jensen's inequality yields that $\left(\mathbb{E}[Z_i^k]\right)^{2l/k} \leq \mathbb{E}[Z_i^{2l}] \leq R$ for all $i = 1, \ldots, n$. This shows that

$$\mathbb{E}[Z_i^k] \leq R^{k/(2l)} \text{ for all } i = 1, \ldots, n. \tag{5.2}$$

Observe that by expanding S_n^{2l} into products of $Z_1, \ldots, Z_n$ and then taking expectations, the terms that involve $l+1$ or more distinct Z_i's will vanish (because $Z_1, \ldots, Z_n$ are independent and $\mathbb{E}[Z_i] = 0$). Now fix $m \leq l$ and consider the sum of the terms involving m distinct Z_i's in the expansion of S_n^{2l} (denote this sum by $S(m)$). Note that the expectation of any term that involves m different Z_i's is less than R. This follows from (5.2), the independence of the Z_i's, and the fact that the sum of the powers of the different Z_i's in this term equals $2l$. Hence, to bound $\mathbb{E}[S(m)]$, it suffices to note that the number of terms in $S(m)$ is bounded from above by $\binom{n}{m} m^{2l}$. This can be seen from the following argument. The number of different combinations involving m distinct Z_i's is $\binom{n}{m}$ and the number of terms in $S(m)$ involving only m particular Z_i's is bounded by m^{2l} (i.e., the number of ways one can sample $2l$ observations from the m particular Z_i's with repetition). Combining this information, we obtain that $\mathbb{E}[S(m)] \leq \binom{n}{m} m^{2l} R \leq n^m m^{2l} R$. Since $m \leq l$, we now obtain that

$$\mathbb{E}[S_n^{2l}] = \sum_{m=1}^{l} \mathbb{E}[S(m)] \leq n^l l^{2l+1} R. \tag{5.3}$$

Combining (5.1) and (5.3) gives the desired result. ∎

We need the following definitions and assumptions. For each $\epsilon > 0$ and $k \in \mathbb{N}$, let

$$B_k(\epsilon) = \{\exists \theta \in \Theta_k \text{ s.t. } f(\theta) \geq f^* - \epsilon\}.$$

Let $\{\mathcal{F}_k\}$ be any filtration such that $B_k(\epsilon) \in \mathcal{F}_k$ for all $\epsilon > 0$ and $k \in \mathbb{N}$. For $n \in \mathbb{N}$ and $\theta \in \Theta$, let $f_n(\theta)$ be the objective function estimate of $f(\theta)$ obtained from n observations

of $f(\theta)$. Finally, let $\bar{A}$ denote the complement of any set A and *i.o.* stand for "infinitely often."

Definition 5.1. A sequence $\{a_k\}$ is said to be $O(k^n)$ for some $n \in \mathbb{R}$ if there exists a $C_1 \in \mathbb{R}^+$ such that $0 \leq a_k \leq C_1 k^n$ for all $k \in \mathbb{N}$. A sequence $\{a_k\}$ is said to be $\Phi(k^n)$ for some $n \in \mathbb{R}$ if there exists a $C_2 \in \mathbb{R}^+$ such that $a_k \geq C_2 k^n$ for all $k \in \mathbb{N}$. A sequence $\{a_k\}$ is said to be $\Omega(k^n)$ for some $n \in \mathbb{R}$ if it is both $O(k^n)$ and $\Phi(k^n)$.

Assumption 5.1. *For each $\theta \in \Theta$, we can generate independent and unbiased observations $\{h_\theta(X_\theta^i)\}_{i=1}^\infty$ of $f(\theta)$. Moreover, there exist $l \in \mathbb{N} \setminus \{0,1\}$ and $R \in \mathbb{R}^+$ such that $\mathbb{E}[(h_\theta(X_\theta^i) - f(\theta))^{2l}] \leq R$ for all $\theta \in \Theta$ and $i \in \mathbb{N}^+$.*

Assumption 5.2. *For each $\epsilon > 0$, $\mathbb{P}(\bar{B}_k(\epsilon)) \to 0$ as $k \to \infty$.*

Assumption 5.3. *For each $\epsilon > 0$, $\sum_{k=1}^\infty \mathbb{P}(\bar{B}_k(\epsilon)|\mathcal{F}_{k-1}) < \infty$ almost surely.*

Observe that Assumption 5.3 implies Assumption 5.2. We now present the convergence analysis of the ASR method and subsequently discuss the required conditions.

Theorem 5.1. *Suppose that $K(i) = \Phi(i^c)$ for some $c > 0$ and that Assumption 5.1 holds. If Assumption 5.2 holds and $c > 1/(l-1)$, then $f(\theta_k^*) \to f^*$ in probability as $k \to \infty$. If Assumption 5.3 holds and $c > (b+1)/(l-1)$, then $f(\theta_k^*) \to f^*$ almost surely as $k \to \infty$.*

Proof: Fix $\epsilon > 0$. First, observe that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon) \leq \mathbb{P}(\bar{B}_k(\epsilon/3)) + \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3)). \tag{5.4}$$

For each $k \in \mathbb{N}$, let $m_k = \lfloor k^{1/b} \rfloor$ and $\tilde{\Theta}_k$ be the set of sampled points by the end of iteration k. Note that m_k is the number of points sampled by the end of iteration k and $\Theta_k \subset \tilde{\Theta}_k$ for all $k \in \mathbb{N}$. Also, suppose that if a sampled point is rejected, we still collect additional observations at this point to ensure that it has enough observations collected at it (i.e., by the end of iteration k it has at least $K(m_k)$ observations). Although we collect additional observations at the points in $\tilde{\Theta}_k \setminus \Theta_k$, we do not use them for making decisions concerning the evolution of the algorithm. This construct is made purely for simplifying the convergence

analysis, and in practice we do not suggest obtaining additional observations at rejected points. Also, for each $i \in \mathbb{N}$, let F_i be the law of θ_i. Then we have that

$$\begin{aligned}
\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3)) &\leq \mathbb{P}\left(\bigcup_{\theta \in \Theta_k} \{|\hat{f}_k(\theta) - f(\theta)| \geq \epsilon/3\} \right) \\
&\leq \mathbb{P}\left(\bigcup_{\theta \in \tilde{\Theta}_k} \{|\hat{f}_k(\theta) - f(\theta)| \geq \epsilon/3\} \right) \\
&= \int_\Theta \mathbb{P}\left(\bigcup_{\theta \in \tilde{\Theta}_k} \{|\hat{f}_k(\theta) - f(\theta)| \geq \epsilon/3\} \middle| \theta_1 = x_1 \right) F_1(\mathrm{d}x_1) \\
&\leq \int_\Theta \mathbb{P}\left(|\hat{f}_k(x_1) - f(x_1)| \geq \epsilon/3 \middle| \theta_1 = x_1 \right) F_1(\mathrm{d}x_1) \\
&\quad + \mathbb{P}\left(\bigcup_{\theta \in \tilde{\Theta}_k \setminus \{\theta_1\}} \{|\hat{f}_k(\theta) - f(\theta)| \geq \epsilon/3\} \right).
\end{aligned} \tag{5.5}$$

Proceeding recursively, we obtain that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3)) \leq \sum_{i=1}^{m_k} \int_\Theta \mathbb{P}\left(|\hat{f}_k(x_i) - f(x_i)| \geq \epsilon/3 \middle| \theta_i = x_i \right) F_i(\mathrm{d}x_i). \tag{5.6}$$

Recall that the number of observations collected at each sampled point by the end of iteration k is at least $K(m_k)$. Thus, for each $i = 1, \ldots, m_k$, we have that

$$\begin{aligned}
\mathbb{P}\left(|\hat{f}_k(x_i) - f(x_i)| \geq \epsilon/3 | \theta_i = x_i \right) &= \sum_{n=K(m_k)}^{\infty} \mathbb{P}\left(|\hat{f}_k(x_i) - f(x_i)| \geq \epsilon/3, N_k(x_i) = n | \theta_i = x_i \right) \\
&= \sum_{n=K(m_k)}^{\infty} \mathbb{P}(|f_n(x_i) - f(x_i)| \geq \epsilon/3, N_k(x_i) = n | \theta_i = x_i) \\
&\leq \sum_{n=K(m_k)}^{\infty} \mathbb{P}(|f_n(x_i) - f(x_i)| \geq \epsilon/3 | \theta_i = x_i) \\
&= \sum_{n=K(m_k)}^{\infty} \mathbb{P}(|f_n(x_i) - f(x_i)| \geq \epsilon/3).
\end{aligned} \tag{5.7}$$

The last equality holds because the objective function observations collected at x_i are independent of the fact that $\theta_i = x_i$.

Combining equations (5.6) and (5.7) yields that for all k sufficiently large,

$$\begin{aligned}
\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3)) &\leq \sum_{i=1}^{m_k} \int_\Theta \sum_{n=K(m_k)}^{\infty} \frac{C_1}{n^l} F_i(\mathrm{d}x_i) \leq \sum_{i=1}^{m_k} \int_\Theta \int_{C_2 k^{c/b}}^{\infty} \frac{C_1}{z^l} \, \mathrm{d}z \, F_i(\mathrm{d}x_i) \\
&= \sum_{i=1}^{m_k} \int_\Theta \frac{C_3}{k^{c(l-1)/b}} F_i(\mathrm{d}x_i) = \frac{C_3 m_k}{k^{c(l-1)/b}} \\
&\leq \frac{C_3}{k^{(c(l-1)-1)/b}},
\end{aligned} \tag{5.8}$$

where C_1, C_2, and C_3 are positive constants. The first inequality follows from Assumption 5.1 and Lemma 5.1. The second inequality follows by approximating the sum by the integral, which is possible because the summand is monotonically decreasing in n, $K(i) = \Phi(i^c)$, and for all $C > 0$, there exists $C_2 > 0$ such that $K(m_k) - 1 \geq Cm_k^c - 1 \geq C_2 k^{c/b} > 0$ for all k large enough. The last inequality holds because $m_k \leq k^{1/b}$.

Hence, if $c > 1/(l-1)$, then $(c(l-1)-1)/b > 0$. From equation (5.8) we obtain that $\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3)) \to 0$ as $k \to \infty$. By Assumption 5.2, we know that $\mathbb{P}(\bar{B}_k(\epsilon/3)) \to 0$ as $k \to \infty$. Equation (5.4) now implies that $\mathbb{P}(f(\theta_k^*) < f^* - \epsilon) \to 0$ as $k \to \infty$. This proves the first assertion of the theorem since ϵ is arbitrary.

We now prove the second assertion. First, observe that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon | \mathcal{F}_{k-1}) \leq \mathbb{P}(\bar{B}_k(\epsilon/3) | \mathcal{F}_{k-1}) + \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3) | \mathcal{F}_{k-1}). \quad (5.9)$$

Note that if $c > (b+1)/(l-1)$, then $(c(l-1)-1)/b > 1$. From equation (5.8), we obtain that $\sum_{k=1}^{\infty} \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3)) < \infty$. Hence, the random variable $\sum_{k=1}^{\infty} \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/3) | \mathcal{F}_{k-1})$ is almost surely finite because it has finite expectation (the exchange of the order of summation and expectation is justified by Fubini's theorem). Assumption 5.3 and equation (5.9) now give that $\sum_{k=1}^{\infty} \mathbb{P}(f(\theta_k^*) < f^* - \epsilon | \mathcal{F}_{k-1}) < \infty$ with probability one. Thus, by Corollary 2.3 in Hall and Heyde [43], we have that $\{f(\theta_k^*) < f^* - \epsilon \text{ i.o.}\}$ with probability zero. The result now follows from Theorem 4.2.2 in Chung [28] since ϵ is arbitrary. ■

We next briefly comment on Theorem 5.1. First, there are no assumptions made on the resampling strategy. Hence this component of ASR can be controlled adaptively by the end user with the goal of achieving improved empirical performance without affecting the asymptotic convergence guarantee. Second, Assumption 5.2 or 5.3 is the only restriction imposed on the sampling strategy and acceptance criterion. Consequently, ASR is convergent even when the sampling strategy includes some kind of local search, as long as Assumption 5.2 or 5.3 is satisfied. This is one of the major contributions of our work because improving upon good solutions using only global search can be very slow. Third, larger values of l for which Assumption 5.1 holds yield slower growth in the number of objective function

observations at each accepted point required for the method to converge. Also, for a fixed l, a higher growth rate in the number of objective function observations at each accepted point is required for almost sure convergence, as opposed to for convergence in probability (this is reasonable because the former mode of convergence implies the latter). Finally, if l is large, then Theorem 5.1 imposes only mild restrictions on c, and hence on the growth rate of $K(i)$.

Note that the objective function observations collected during the sampling stages of the search (i.e., at iterations when $k = M(i)$) can be viewed as mandatory sampling (because we require that each sampled and accepted point has a sufficient number of observations collected at it), while observations collected during the resampling stages of the search can be viewed as flexible sampling (because we require no conditions on how points are resampled). In practice, it seems desirable for most of the simulation effort to be spent on flexible sampling, which is under the control of the user and hence can be geared toward improving the empirical performance of the method. Observe that the number of mandatory objective function observations collected by the end of iteration k is at most $K(m_k) \times m_k$ because we sample and accept at most m_k points and at each point we need to collect at most $K(m_k)$ observations. Suppose that $K(i)$ is a $\Omega(i^c)$ sequence. Then we have that

$$\frac{K(m_k) \times m_k}{k} = \Omega(k^{(c+1-b)/b}).$$

Thus, if $c+1 < b$ (which obviously requires that $b > 1$), then asymptotically the number of observations collected during the (mandatory) sampling stages is sublinear, and hence ASR will at least eventually spend most of the simulation effort during the (flexible) resampling stages of the search (because the number of observations collected by ASR by the end of iteration k is $\Phi(k)$). Note that when Assumption 5.1 holds (so that $l \geq 2$), it is always possible to pick b and c so that the ASR algorithm is not only convergent in probability (we need $l \geq 3$ for almost surely) but also asymptotically spends most of the simulation effort during the resampling stages. On the other hand, if $c+1 \geq b$, then ASR may or may not spend most of the simulation effort during the resampling stages, depending on how conservative the above analysis is.

5.2.3 Discussion of Assumption 5.3

In this section we discuss how Assumption 5.3 can be satisfied in practice. This is the key assumption required for guaranteeing the convergence of the ASR approach, see Theorem 5.1 in Section 5.2.2. Recall that Assumption 5.3 implies Assumption 5.2.

Suppose that the sequence $\{\theta_i\}$ of points sampled by ASR is a deterministic sequence that is dense in Θ, that f is continuous, and that every sampled point is accepted. Then, it is easy to see that Assumption 5.3 is satisfied. In the case where Θ is bounded, for instance, $\{\theta_i\}$ can be a low-discrepancy sequence (see Niederreiter [67] for more details).

We next consider another method that satisfies this assumption. The sampling procedure in this case is as follows. With probability $g > 0$, a new solution θ is sampled independently of everything else from a distribution G, and with probability $1 - g$, a new solution θ is sampled based on some adaptive sampling procedure (e.g., a local search procedure). The first component is intended for exploring the entire feasible region in order to identify "good" solutions. The second component aims at exploitation (or local search) of regions that contain "good" solutions, and can be specified arbitrarily by the user without affecting the convergence guarantee. This component can be adaptive in the sense that it may depend on all the information gathered by the search method so far, a feature that can be especially useful from the perspective of empirical performance. We have also used similar ideas in Chapter 3. The acceptance criterion is such that every sampled point is accepted.

For each $\epsilon > 0$, let $\Theta_\epsilon = \{\theta \in \Theta : f(\theta) \geq f^* - \epsilon\}$. Note that $B_k(\epsilon) = \{\Theta_k \cap \Theta_\epsilon \neq \emptyset\}$. We need the following assumption.

Assumption 5.4. *For each* $\epsilon > 0$, $G(\Theta_\epsilon) > 0$.

We have the following proposition.

Proposition 5.1. Suppose that Assumption 5.4 holds. Then, Assumption 5.3 is satisfied by the method described above.

Proof: Let $\mathcal{F}_k$ be a filtration such that $B_k(\epsilon) \in \mathcal{F}_k$ for all $\epsilon > 0$ and $k \in \mathbb{N}$. Fix $\epsilon > 0$. By

the monotone convergence theorem, we have that

$$\mathbb{E}\left[\sum_{k=1}^{\infty}\mathbb{P}(\bar{B}_k(\epsilon)|\mathcal{F}_{k-1})\right] = \sum_{k=1}^{\infty}\mathbb{E}\left[\mathbb{P}(\bar{B}_k(\epsilon)|\mathcal{F}_{k-1})\right] = \sum_{k=1}^{\infty}\mathbb{P}(\bar{B}_k(\epsilon)).$$

Hence, it suffices to verify that $\sum_{k=1}^{\infty}\mathbb{P}(\bar{B}_k(\epsilon)) < \infty$. Recall that the number of points sampled by iteration k is $m_k = \lfloor k^{1/b} \rfloor$. Note also that the probability of sampling a point in Θ_ϵ in any iteration $M(i)$ is at least $g \times G(\Theta_\epsilon)$. Hence we have that $\mathbb{P}(\bar{B}_k(\epsilon)) \leq (1 - g \times G(\Theta_\epsilon))^{m_k}$. Assumption 5.4 and the fact that $g > 0$ imply that $1 - g \times G(\Theta_\epsilon) < 1$. Without loss of generality, we can assume that $1 - g \times G(\Theta_\epsilon) > 0$ (because we are done otherwise). Hence, let $\lambda > 0$ be such that $\exp(-\lambda) = 1 - g \times G(\Theta_\epsilon)$. We have that

$$\sum_{k=1}^{\infty}\mathbb{P}(\bar{B}_k(\epsilon)) \leq \sum_{k=1}^{\infty}\exp(-\lambda(k^{1/b} - 1)) \leq \exp(\lambda)\int_0^{\infty}\exp(-\lambda x^{1/b})\,\mathrm{d}x < \infty.$$

The second inequality follows by the fact that $\exp(-\lambda k^{1/b})$ is a monotonically decreasing function of k. The last inequality follows from the change of variable $y = x^{1/b}$ and the facts that $b \geq 1$ and an exponential random variable with rate $\lambda > 0$ has all moments finite. This completes the proof. ■

We now consider a third method that satisfies Assumption 5.3. The sampling procedure is as described in the previous method. We next describe the acceptance criterion. Let $\delta > 0$. The newly sampled solution is included in the set Θ_k of sampled and accepted points in iteration k if an objective function estimate based on $K \geq 1$ observations at this point is at least as good as the estimated objective function value at the best solution found so far minus an indifference parameter δ (i.e., a sampled point θ is accepted if $f_K(\theta) \geq \hat{f}_{k-1}(\theta^*_{k-1}) - \delta$). The idea is that the sampled point is only accepted if there is sufficient indication that it might have a higher objective function value than the best point found so far. We assume that the first sampled point is always accepted. We now present conditions under which Assumption 5.3 is satisfied for this method. We need the following assumption.

Assumption 5.5. *There exists $\bar{\epsilon} > 0$ such that $\int_{\Theta_\epsilon}\mathbb{P}(f_K(\theta) \geq f^* + \epsilon - \delta)\,G(\mathrm{d}\theta) > 0$ for all $\epsilon \in (0, \bar{\epsilon}]$.*

Assumption 5.5 is satisfied, for instance, if $\bar{\epsilon} = \delta/2$, Assumption 5.4 holds, and $\inf_{\theta\in\Theta_{\delta/2}}\mathbb{P}(f_K(\theta) \geq f(\theta)) > 0$. The latter assumption is satisfied, for example, if the

distribution of $f_K(\theta)$ is symmetric around $f(\theta)$ for each $\theta \in \Theta_{\delta/2}$. We are now ready to prove the following result.

Proposition 5.2. Suppose that Assumptions 5.1, 5.4, and 5.5 hold and $K(i) = \Phi(i^c)$ for some $c > 2/(l-1)$, where $l \geq 2$. Then, Assumption 5.3 is satisfied by the method described above.

Proof: For each $k \in \mathbb{N}$, let $\mathcal{F}_k$ be the history generated by the ASR method by the end of iteration k. Observe that it suffices to prove the assertion of the proposition for $\epsilon \in (0, \bar{\epsilon}]$. Hence, fix $\epsilon \in (0, \bar{\epsilon}]$. For each $i \in \mathbb{N}^+$, let $D_{M(i)}$ be the event that a point in Θ_ϵ is sampled and also accepted in iteration $M(i)$. Note that $D_{M(i)}$ is measurable with respect to $\mathcal{F}_{M(i)}$. Fix $i \in \mathbb{N}^+$ and let $A_{M(i)}$ be the event that sampling of a new point is made using the distribution G in iteration $M(i)$. Let $I(\cdot)$ denote an indicator function and note that

$$\begin{aligned} I(D_{M(i)}) &= I(\theta_i \in \Theta_\epsilon) I(f_K(\theta_i) \geq \hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) - \delta) \\ &\geq I(\theta_i \in \Theta_\epsilon) I(A_{M(i)}) I(f_K(\theta_i) \geq f^* + \epsilon - \delta) I(\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) \leq f^* + \epsilon). \end{aligned}$$

Furthermore, note that $\{\theta_i \in \Theta_\epsilon, A_{M(i)}, f_K(\theta_i) \geq f^* + \epsilon - \delta\}$ does not depend on the past history $\mathcal{F}_{M(i)-1}$, and that $\{\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) \leq f^* + \epsilon\}$ is measurable with respect to $\mathcal{F}_{M(i)-1}$. Hence, taking conditional expectation yields

$$\begin{aligned} \mathbb{P}(D_{M(i)}|\mathcal{F}_{M(i)-1}) &\geq I(\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) \leq f^* + \epsilon) \mathbb{P}(A_{M(i)}) \mathbb{P}(\theta_i \in \Theta_\epsilon | A_{M(i)}) \\ &\quad \times \mathbb{P}(f_K(\theta_i) \geq f^* + \epsilon - \delta | A_{M(i)}, \theta_i \in \Theta_\epsilon). \end{aligned} \tag{5.10}$$

Note that $\mathbb{P}(A_{M(i)}) = g > 0$ and $\mathbb{P}(\theta_i \in \Theta_\epsilon | A_{M(i)}) = G(\Theta_\epsilon) > 0$ by Assumption 5.4. Moreover, because objective function observations are collected independently of everything else, by conditioning on θ_i (note that we are given that sampling is made using G) we obtain that

$$\mathbb{P}(f_K(\theta_i) \geq f^* + \epsilon - \delta | A_{M(i)}, \theta_i \in \Theta_\epsilon) = \frac{1}{G(\Theta_\epsilon)} \int_{\Theta_\epsilon} \mathbb{P}(f_K(\theta) \geq f^* + \epsilon - \delta)\, G(\mathrm{d}\theta) > 0.$$

The last inequality follows from Assumptions 5.4 and 5.5. Equation (5.10) now yields that

$$\mathbb{P}(D_{M(i)}|\mathcal{F}_{M(i)-1}) \geq C_1 I(\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) \leq f^* + \epsilon), \tag{5.11}$$

where $C_1 > 0$.

We now verify that $\{\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) > f^* + \epsilon \text{ i.o.}\}$ with probability zero. First, note that

$$\{\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) > f^* + \epsilon\} \subset \bigcup_{\theta \in \Theta_{M(i)-1}} \{|\hat{f}_{M(i)-1}(\theta) - f(\theta)| \geq \epsilon\}.$$

Proceeding similarly as in equations (5.5) through (5.8), for $i \geq 3$, we obtain that

$$\mathbb{P}(\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) > f^* + \epsilon) \leq \frac{C_2}{(M(i)-1)^{(c(l-1)-1)/b}} \leq \frac{C_2}{(i-2)^{(c(l-1)-1)}},$$

where $C_2 > 0$. Note that the proof of this conclusion does not rely on Assumption 5.3. The second inequality holds because for $i \geq 3$ and $b \geq 1$, we have $M(i) - 1 = \lfloor i^b \rfloor - 1 \geq i^b - 2 \geq (i-2)^b$, where the last inequality holds because x^b is convex on $[0, \infty)$ and differentiable. Because $c > 2/(l-1)$, we now obtain that $\sum_{i=1}^{\infty} \mathbb{P}(\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) > f^* + \epsilon) < \infty$. The first Borel-Cantelli lemma now implies that $\{\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) > f^* + \epsilon \text{ i.o.}\}$ with probability zero.

Let $\Omega_1 = \{\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) > f^* + \epsilon \text{ i.o.}\}$. For every $\omega \in \bar{\Omega}_1$, there exists i' large enough so that $\hat{f}_{M(i)-1}(\theta^*_{M(i)-1}) \leq f^* + \epsilon$ for all $i \geq i'$. Equation (5.11) now yields that for all $i \geq i'$, we have $\mathbb{P}(D_{M(i)}|\mathcal{F}_{M(i)-1})(\omega) \geq C_1$, and hence $\sum_{i=1}^{\infty} \mathbb{P}(D_{M(i)}|\mathcal{F}_{M(i)-1})(\omega) = \infty$. By Corollary 2.3 in Hall and Heyde [43], we obtain that $\{D_{M(i)} \text{ i.o.}\}$ with probability one because $\mathbb{P}(\bar{\Omega}_1) = 1$. Let $\Omega_2 = \{D_{M(i)} \text{ i.o.}\}$. Recall that the number of points sampled by iteration k (but not necessarily accepted) is $m_k = \lfloor k^{1/b} \rfloor$, and hence $I(\bar{B}_k(\epsilon)) = \prod_{i=1}^{m_k} I(\bar{D}_{M(i)}) \leq \prod_{i=1}^{m_{k-1}} I(\bar{D}_{M(i)})$. Note that $i \leq m_{k-1}$ if and only if $M(i) \leq k-1$. Since $D_{M(i)} \in \mathcal{F}_{M(i)}$, we have that $\bar{D}_{M(i)} \in \mathcal{F}_{k-1}$ for $i \leq m_{k-1}$, and consequently that

$$\mathbb{P}(\bar{B}_k(\epsilon)|\mathcal{F}_{k-1}) \leq \mathbb{E}\left[\prod_{i=1}^{m_{k-1}} I(\bar{D}_{M(i)}) \Big| \mathcal{F}_{k-1}\right] = \prod_{i=1}^{m_{k-1}} I(\bar{D}_{M(i)}).$$

Fix $\omega \in \Omega_2$. Then, there exists i large enough such that $I(D_{M(i)})(\omega) = 1$, and so for all k large enough we get that $\mathbb{P}(\bar{B}_k(\epsilon)|\mathcal{F}_{k-1})(\omega) = 0$ and hence

$$\sum_{k=1}^{\infty} \mathbb{P}(\bar{B}_k(\epsilon)|\mathcal{F}_{k-1})(\omega) < \infty.$$

This concludes the proof because $\mathbb{P}(\Omega_2) = 1$. ∎

Note that the condition on the value of c in Proposition 5.2 is not very restrictive if we desire ASR to be almost surely convergent (because for almost sure convergence in Theorem 5.1 we require that $c > (b+1)/(l-1)$ and $b \geq 1$) or if the value of l is large.

5.3 Deterministic Shrinking Ball Algorithm

Before presenting the second approach to solve the problem (1.1), we need to make a few definitions. For each $\theta \in \Theta$ and $r \in [0, \infty)$, let $B(\theta, r) = \{\theta' \in \Theta : d(\theta, \theta') \leq r\}$. Let G be a distribution on the feasible region Θ. For each subset A of Θ, let $N_k(A)$ be the number of points sampled in A by the end of iteration k. Let $\{r_k\}$ be a deterministic sequence of positive real numbers. For each $k \in \mathbb{N}$ and $\theta \in \Theta$, let $S'_k(\theta)$ be the sum of the $N_k(B(\theta, r_k))$ objective function observations collected at the points in $B(\theta, r_k)$ by iteration k. Let Θ_k be the set of points sampled by the end of iteration k. Observe that $N_k(\cdot)$, $S'_k(\cdot)$, and Θ_k are stochastic. More formally the second method is stated in Algorithm 5.2. This method was first introduced and analyzed by Baumert and Smith [20].

Algorithm 5.2 Deterministic Shrinking Ball (DSB) Algorithm

1: Select a sequence $\{r_k\}$ and the global sampling distribution G. Let $k = 0$ and $\Theta_0 = \emptyset$.
2: **while** Stopping criterion is not satisfied **do**
3: Let $k = k + 1$
4: Sample a solution θ_k from the global distribution G independent of everything else
5: Let $\Theta_k = \Theta_{k-1} \cup \{\theta_k\}$
6: Obtain an estimate of the objective function value at θ independent of everything else
7: **end while**
8: For each $\theta \in \Theta_k$, compute $S'_k(\theta)$ and $N_k(B(\theta, r_k))$
9: Select an estimate of the optimal solution

$$\theta_k^* \in \arg\max_{\theta \in \Theta_k} \hat{f}_k(\theta) = \frac{S'_k(\theta)}{N_k(B(\theta, r_k))}$$

We now briefly describe the DSB approach. At each iteration we sample a new solution from the distribution G and obtain an objective function estimate at the sampled solution, independent of everything else. The estimate of the objective function value at any feasible point θ at the end of iteration k is the average of the objective function observations collected at points that are at most r_k distance units away from θ. The estimate θ_k^* of the optimal

solution in iteration k is an already sampled point that has the highest estimated objective function value. Also, notice that if $S'_k(\theta)$ and $N_k(B(\theta, r_k))$, where $\theta \in \Theta_k$, are not used by the stopping criterion, then we need to calculate these only once when search is terminated. Hence, for empirical efficiency, we suggest calculating $S'_k(\theta)$ and $N_k(B(\theta, r_k))$ for each $\theta \in \Theta_k$ (and hence θ^*_k) only when the search is terminated. Also, note the generality of the definition of $B(\theta, r)$ in the sense that if $\Theta \subset \mathbb{R}^s$ and d is a Euclidean norm on $\mathbb{R}^s$, then $B(\theta, r)$ is an s-dimensional ball. On the other hand, if $\Theta \subset \mathbb{R}^s$ and d is a sup norm on $\mathbb{R}^s$, then $B(\theta, r)$ is an s-dimensional cube. Hence the estimated objective function value at each sampled point can be computed, for instance, as an average of the objective function estimates of points in either an enclosing ball or cube (see Assumption 5.7 and Remark 5.2 below).

We now briefly comment on the choice of the sequence $\{r_k\}$. For the convergence of the method, we require that $r_k \to 0$ as $k \to \infty$ (see Theorem 5.2 below). Note that if r_k decreases to 0 rapidly (slowly), then the bias in the estimated objective function value at each point will be low (high), but the variance of this estimate will be high (low) because we average a smaller (larger) number of the objective function observations. Thus, there is a tradeoff in deciding how rapidly r_k should decrease. A user of the DSB algorithm should strive to achieve a good balance between these effects to ensure good empirical performance of the method.

Our convergence analysis of DSB relies extensively on the ideas of Baumert and Smith [20]. The main contributions of our theoretical analysis of the method are as follows. First, we assume a more general form of the noise structure in the objective function estimates. This allows us to prove the convergence of the method both in probability and almost surely, while Baumert and Smith [20] show that their method is convergent in probability. Moreover, our analysis explicitly identifies the relationship between the rates at which r_k can decrease and the noise in the objective function estimates, so that the convergence of the method (either in probability or almost surely) is guaranteed.

We now analyze the convergence of the DSB method. We first present two assumptions.

Assumption 5.6. *$\Theta = \cup_{i=1}^{n} \Theta_i$ and G is the uniform distribution on Θ, where for each $i = 1, \ldots, n$, Θ_i is a convex and bounded subset of $\mathbb{R}^s$ such that $G(\Theta_i) > 0$.*

Assumption 5.7. *d is a Euclidean distance metric on $\Theta \subset \mathbb{R}^s$.*

These assumptions are of crucial importance in the proof of the convergence of DSB. They enable us to show that eventually every sampled solution has a "sufficient" number of sampled points within a certain distance to it, even though this distance decreases as the number of iterations grows. The interested reader is referred to Baumert and Smith [20] for more details on Assumption 5.6 and to Remark 5.1 (provided at the end of this section) for a discussion on how the assumption about the distribution G can be relaxed. We will also need the following lemma. The proof of this lemma resembles the proof of Lemma 2.6 in Baumert and Smith [20] and is provided in Appendix C.

Lemma 5.2. *Let $p, q \in (0,1)$ be such that $p + q < 1$. Suppose that Assumptions 5.6 and 5.7 hold, let $r_k = \Phi(k^{-p/s})$ be a sequence of positive real numbers such that $r_k \to 0$ as $k \to \infty$, and $L_k = O(k^q)$ be a sequence of positive integers. For each $k \in \mathbb{N}$, let $D_k = \{\forall \theta \in \Theta, N_k(B(\theta, r_k)) \geq L_k\}$. Then $\sum_{k=1}^{\infty} \mathbb{P}(\bar{D}_k) < \infty$.*

We next state and prove our main convergence result concerning the DSB method.

Theorem 5.2. *Suppose that Assumptions 5.1, 5.4, 5.6, and 5.7 are satisfied, $r_k = \Phi(k^{-p/s})$ with $p > 0$, $r_k \to 0$ as $k \to \infty$, and f is uniformly continuous. If $p < 1 - 1/l$, then $f(\theta_k^*) \to f^*$ in probability as $k \to \infty$. If $p < 1 - 2/l$, then $f(\theta_k^*) \to f^*$ almost surely as $k \to \infty$.*

Proof: Fix $\epsilon > 0$. Let L_k be a $\Omega(k^q)$ function, where $q \in (0,1)$ and $p + q < 1$. Also, let C_k be the event that $N(B(\theta, r_k)) \geq L_k$ for all $\theta \in \Theta_k$. Observe that for each $k \in \mathbb{N}$, we have that $D_k \subset C_k$. Then we have that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon) \leq \mathbb{P}(\bar{B}_k(\epsilon/5)) + \mathbb{P}(\bar{D}_k) + \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), C_k). \tag{5.12}$$

Note that $\mathbb{P}(\bar{B}_k(\epsilon/5)) = (1 - G(\Theta_{\epsilon/5}))^k$. Hence, Assumption 5.4 ensures that

$$\sum_{k=1}^{\infty} \mathbb{P}(\bar{B}_k(\epsilon/5)) < \infty. \tag{5.13}$$

Because f is uniformly continuous, there exists $\delta > 0$ such that $d(\theta_1, \theta_2) \leq \delta$ implies that $|f(\theta_1) - f(\theta_2)| \leq \epsilon/5$. Since $r_k \to 0$ as $k \to \infty$ and $L_k = \Omega(k^q)$, there exists k' large enough

so that $r_k \leq \delta$ and $L_k \geq Lk^q$ for $k \geq k'$. Fix $k \geq k'$. Let A be a set of k deterministic points in Θ and suppose that $B_k(\epsilon/5)$ and C_k occur when $\Theta_k = A$. Then we have that

$$\begin{aligned}\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), C_k | \Theta_k = A) &\leq \mathbb{P}(\cup_{\theta \in \Theta_k} \{|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5\} | \Theta_k = A) \\ &\leq \sum_{\theta \in A} \mathbb{P}(|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5 | \Theta_k = A). \quad (5.14)\end{aligned}$$

The first inequality follows because the event $\{f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5)\}$ can only happen if $|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5$ for some $\theta \in \Theta_k$. Now, for each $\theta \in \Theta_k = A$, there exists a sequence of points $\{x_i\}_{i=1}^{n_k(\theta)}$ in A that are within a distance of r_k to θ, with both $n_k(\theta)$ and $x_1, \ldots, x_{n_k(\theta)}$ being deterministic given $\Theta_k = A$ (we omit the dependency of $x_1, \ldots, x_{n_k(\theta)}$ on k and θ for notational simplicity). Thus, we obtain that for all $\theta \in A$,

$$\begin{aligned}&\mathbb{P}(|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5 | \Theta_k = A) \\ &= \mathbb{P}\left(\left| \sum_{i=1}^{n_k(\theta)} h_{x_i}(X_{x_i}) - \sum_{i=1}^{n_k(\theta)} f(x_i) + \sum_{i=1}^{n_k(\theta)} f(x_i) - n_k(\theta) f(\theta) \right| > 2\epsilon n_k(\theta)/5 \middle| \Theta_k = A \right) \\ &\leq \mathbb{P}\left(\left| \sum_{i=1}^{n_k(\theta)} h_{x_i}(X_{x_i}) - \sum_{i=1}^{n_k(\theta)} f(x_i) \right| + \left| \sum_{i=1}^{n_k(\theta)} f(x_i) - n_k(\theta) f(\theta) \right| > 2\epsilon n_k(\theta)/5 \middle| \Theta_k = A \right) \\ &\leq \mathbb{P}\left(\left| \sum_{i=1}^{n_k(\theta)} h_{x_i}(X_{x_i}) - \sum_{i=1}^{n_k(\theta)} f(x_i) \right| \geq \epsilon n_k(\theta)/5 \middle| \Theta_k = A \right) \\ &= \mathbb{P}\left(\left| \sum_{i=1}^{n_k(\theta)} h_{x_i}(X_{x_i}) - \sum_{i=1}^{n_k(\theta)} f(x_i) \right| \geq \epsilon n_k(\theta)/5 \right) \\ &\leq \frac{C}{(n_k(\theta))^l} \\ &\leq \frac{C}{(Lk^q)^l}. \quad (5.15)\end{aligned}$$

The first inequality holds by the triangular inequality, while the second inequality follows from the fact that f is uniformly continuous and $r_k \leq \delta$. The second equality follows from the fact that the estimates of the objective function values do not depend on the sampled points, and hence conditioning on Θ_k is redundant (once the values of $n_k(\theta)$ and $x_1, \ldots, x_{n_k(\theta)}$ have been inserted). The third inequality follows from Assumption 5.1 and Lemma 5.1. The final inequality holds because $\Theta_k = A$ is such that C_k occurs (i.e., $n_k(\theta) \geq L_k \geq Lk^q$). The derivation of equation (5.15) above resembles the proof of Lemma 2.7 in Baumert and Smith [20]. Combining equations (5.14) and (5.15) and recalling that $|\Theta_k| = k$,

we obtain that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), C_k | \Theta_k = A) \leq \frac{const}{k^{ql-1}}.$$

Observe that the inequality above holds trivially when $\Theta_k = A$ is such that either $B_k(\epsilon/5)$ or C_k or both do not occur. Hence, unconditioning of the expression above yields

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), C_k) \leq \frac{const}{k^{ql-1}}. \tag{5.16}$$

If $p < 1 - 1/l$, then we can choose $q > 1/l$. This implies that $\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), C_k) \to 0$ as $k \to \infty$. Combining this with equations (5.12) and (5.13) and Lemma 5.2, we get that $f(\theta_k^*) \to f^*$ in probability as $k \to \infty$.

Similarly, if $p < 1 - 2/l$, then we can pick $q > 2/l$. This implies that $\sum_{k=1}^{\infty} \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), C_k) < \infty$. Combining this with equations (5.12) and (5.13), Lemma 5.2, and the first Borel-Cantelli lemma, we get that $\{f(\theta_k^*) < f^* - \epsilon \text{ i.o.}\}$ with probability zero. The result now follows from Theorem 4.2.2 in Chung [28] since ϵ is arbitrary. ■

We now briefly compare Theorems 5.1 and 5.2. The conditions under which the DSB algorithm converges are more restrictive than the ones for ASR (for instance, they do not allow any part of the method to be adaptive and hence do not include a local search component) because it is not easy to show that these methods are convergent under more general conditions. The main difficulty in proving the convergence of an adaptive version of DSB arises in bounding from above the probability that the estimate of the objective function value at each sampled point θ is not close enough to $f(\theta)$ because the objective function observations used in obtaining this estimate may depend on all the points sampled by the algorithm (i.e., in this case it is difficult to derive an equation resembling equation (5.16)). Also, in order for DSB to converge, we impose structural assumptions on the underlying optimization problem, such as the uniform continuity of f and the specific form of Θ (see Assumption 5.6). Such structural assumptions are not needed in Theorem 5.1. Finally, the minimum value for l for which ASR converges either in probability or almost surely is 2, while it is 2 for convergence in probability and 3 for almost sure convergence for DSB. The reasons for the difference in the minimum value of l under which the methods converge almost surely are that by the end of iteration k of the ASR method, we have fewer sampled

and accepted points and also it is possible to collect more than k objective function value observations at these points (and hence have better knowledge about the corresponding objective function values), while by the end of iteration k of the DSB approach, we always sample k points and collect one objective function observation at each of these points.

Also, note that the bounds on p in Theorem 5.2 constrain how rapidly r_k can decrease so that the convergence of the method is still guaranteed. This is consistent with the earlier discussion that r_k should decrease sufficiently slowly to ensure that the effects of noise are not too significant. Moreover, note that, as expected, a weaker condition on p is required for convergence in probability as opposed to almost sure convergence.

Remark 5.1. Observe that the assumption on the global sampling distribution G (see Assumption 5.6) is only in proving Lemma 5.2. From the proof of Lemma 5.2 and Lemmas 2.3 and 2.4 in Baumert and Smith [20], it should be obvious that the conclusion of the lemma will hold if G has a density function $g(\theta)$ such that $g(\theta) \geq \epsilon > 0$ for all $\theta \in \Theta$. Hence, the DSB method is also convergent under the assumptions of Theorem 5.2 with the assumption on G substituted by the aforementioned assumption.

Remark 5.2. Note that Assumption 5.7 is used only in proving Lemma 5.2. From the proof of Lemma 5.2 and Lemmas 2.3 and 2.4 in Baumert and Smith [20], it should be obvious that the conclusion of the lemma will hold with a number of different metrics. For instance, it is still valid with a metric d being a sup norm on $\mathbb{R}^s$. Thus, DSB is also convergent under the assumptions of Theorem 5.2 with Assumption 5.7 substituted by the metric d being a sup norm.

We next briefly discuss implementation issues related to DSB. Consider the following naive implementation of the approach. Suppose that during the search phase of the method the information regarding each sampled point (i.e., its location and its objective function estimate) is stored in a list. Then to compute the estimate of the objective function value at each sampled solution, we need to traverse the list once. Hence the computation of the estimator of the optimal solution requires $O(k^2)$ operations, where k is the iteration number when the search is terminated. This can pose a significant computational overhead when k is large. As a future research direction, it might be desirable to identify a more efficient

method for computing θ_k^*. Another viable research direction would be the identification of efficient heuristics for approximating θ_k^* that produce good empirical results.

5.4 Stochastic Shrinking Ball Algorithm

In this section we introduce our third algorithm. Like DSB, this method is based on pure random search, but it uses a different approach for estimating the objective function values. More specifically, in this section the estimate of the objective function value at any $\theta \in \Theta$ is the average of the objective function value estimates at the n_k closest sampled points to θ. This leads to a different estimator of the optimal solution, which is an already sampled point that has the highest estimated objective function value.

Before stating our approach, we provide the following definitions. For each iteration k and $\theta \in \Theta_k$, let $R_k(\theta)$ be the smallest real number such that $N_k(B(\theta, R_k(\theta))) \geq n_k$. Our algorithm generates one feasible point at a time, and hence we require that $n_k \leq k$ so that $R_k(\theta)$ is well defined. We also let $S_k''(\theta)$ be the sum of the $N_k(B(\theta, R_k(\theta)))$ objective function observations collected at the points in $B(\theta, R_k(\theta))$ by the end of iteration k. Observe that $N_k(\cdot), R_k(\cdot)$, and $S_k''(\cdot)$ are stochastic. As in Section 5.3, note the generality of the definition of d and $B(\theta, r)$. Our third optimization approach is stated in Algorithm 5.3.

Algorithm 5.3 Stochastic Shrinking Ball (SSB) Algorithm

1: Select a sequence $\{n_k\}$ and the global sampling distribution G. Let $k = 0$ and $\Theta_0 = \emptyset$.
2: **while** Stopping criterion is not satisfied **do**
3: Let $k = k + 1$
4: Sample a solution θ_k from the global distribution G independent of everything else
5: Let $\Theta_k = \Theta_{k-1} \cup \{\theta_k\}$
6: Obtain an estimate of the objective function at θ independent of everything else
7: **end while**
8: For each $\theta \in \Theta_k$, compute $S_k''(\theta)$ and $N_k(B(\theta, R_k(\theta)))$
9: Select an estimate of the optimal solution

$$\theta_k^* \in \arg\max_{\theta \in \Theta_k} \hat{f}_k(\theta) = \frac{S_k''(\theta)}{N_k(B(\theta, R_k(\theta)))}$$

We now briefly describe the SSB approach. As in Section 5.3, in each iteration we sample a new solution from the distribution G and obtain a single objective function estimate at that solution, independent of everything else. The estimate of the objective function value at any

feasible point θ at the end of iteration k is the average of the objective function observations collected at the points that are at most $R_k(\theta)$ distance units away from θ. Note that we ensure that the objective function estimate at each sampled point is based on a "sufficient" number of observations through the sequence $\{n_k\}$. The estimate θ_k^* of the optimal solution in iteration k is an already sampled point with the highest estimated objective function value. Also, note that if $S_k''(\theta)$ and $N_k(B(\theta, R_k(\theta)))$, where $\theta \in \Theta_k$, are not used by the stopping criterion, then we need to calculate these only once when search is terminated. Hence, for empirical efficiency, we suggest calculating $S_k''(\theta)$ and $N_k(B(\theta, R_k(\theta)))$ for each $\theta \in \Theta_k$ (and hence θ_k^*) only when the search is terminated. Finally, note that $N_k(\cdot)$ is usually equal to n_k but this need not be the case (e.g., if G has atoms).

We next briefly comment on the choice of the sequence $\{n_k\}$. For the convergence of the method, we require that $n_k \to \infty$ as $k \to \infty$ (see Theorem 5.3 below). Note that if n_k increases rapidly (slowly), then the bias in the estimate of the objective function value at each point will be high (low) but the variance of this estimate will be low (high) because we average a larger (smaller) number of the objective function value observations. Hence, there again exists a tradeoff in determining how fast n_k should increase. A user of SSB should attempt to balance these effects to ensure good empirical performance of the method.

Recall that the number of observations used in obtaining the objective function estimate at each sampled point in DSB is random, whereas the objective function estimate at each sampled point in SSB is an average of at least n_k observations. Due to these properties, we expect that SSB in general will have more stable numerical performance than DSB (i.e., the variation in the quality of the estimator of the optimal solution from replication to replication will be smaller). Thus, if the computational overhead of computing the estimate of the optimal solution is small when compared to the cost of conducting simulations, then SSB may be more desirable than DSB (see Section 5.5.3 for a numerical comparison of the DSB and SSB approaches which shows that these methods have either similar performance or SSB is better than DSB). Moreover, it may be more intuitive to pick the sequence $\{n_k\}$ for SSB in practice, rather than the sequence $\{r_k\}$ for DSB (because the sequence $\{n_k\}$ in the SSB method directly controls the noise in the objective function estimates, while the

sequence $\{r_k\}$ in the DSB method does so implicitly).

We next discuss the convergence of SSB. We will need the following assumption.

Assumption 5.8. $\inf_{\theta\in\Theta} G(B(\theta, r)) > 0$ *for all* $r > 0$.

In light of Lemma 2.4 in Baumert and Smith [20], Assumptions 5.6 and 5.7 imply Assumption 5.8. Assumption 5.8 also holds when Assumptions 5.6 and 5.7 hold with the global sampling distribution G as given in Remark 5.1. We will need the following lemma.

Lemma 5.3. *Suppose that Assumption 5.8 is satisfied and* Θ *contains at least two distinct points. Then there exists* $r' > 0$ *such that for all* $r \leq r'$ *we have that*

$$\sup_{\theta\in\Theta} G(B(\theta, r)) < 1.$$

Proof: Choose $\theta_1, \theta_2 \in \Theta$ with $\theta_1 \neq \theta_2$ so that $d(\theta_1, \theta_2) > 0$. Let $r' = d(\theta_1, \theta_2)/5$. Then, for any $\theta \in \Theta$, we have that either $B(\theta, r') \cap B(\theta_1, r') = \emptyset$ or $B(\theta, r') \cap B(\theta_2, r') = \emptyset$. Hence,

$$G(B(\theta, r')) \leq 1 - \min\{G(B(\theta_1, r')), G(B(\theta_2, r'))\} \leq 1 - \inf_{\theta'\in\Theta} G(B(\theta', r')).$$

Since $r \leq r'$, we now obtain that

$$\sup_{\theta\in\Theta} G(B(\theta, r)) \leq \sup_{\theta\in\Theta} G(B(\theta, r')) \leq 1 - \inf_{\theta'\in\Theta} G(B(\theta', r')) < 1,$$

where the last inequality follows from Assumption 5.8. ■

We are now ready to state and prove our main convergence result concerning SSB.

Theorem 5.3. *Suppose that* $n_k = \Omega(k^q)$*, where* $q \in (0, 1)$ *and* $n_k \leq k$ *for all* $k \in \mathbb{N}^+$*. Also, assume that* f *is uniformly continuous and Assumptions 5.1, 5.4, and 5.8 hold. If* $q > 1/l$*, then* $f(\theta_k^*) \to f^*$ *in probability as* $k \to \infty$*. If* $q > 2/l$*, then* $f(\theta_k^*) \to f^*$ *almost surely as* $k \to \infty$*.*

Proof: Without loss of generality we can assume that $|\Theta| \geq 2$. Fix $\epsilon > 0$. Because f is uniformly continuous, there exists $\delta \in (0, r']$ such that for all $\theta, \theta' \in \Theta$ with $d(\theta, \theta') \leq \delta$, we have that $|f(\theta) - f(\theta')| \leq \epsilon/5$, where r' is defined in Lemma 5.3. Let $\alpha = \max\{1 - \inf_{\theta\in\Theta} G(B(\theta, \delta)), \sup_{\theta\in\Theta} G(B(\theta, \delta))\}$. Assumption 5.8 and Lemma 5.3 ensure that $\alpha \in (0, 1)$. For each $k \in \mathbb{N}$, define $R_k = \sup_{\theta\in\Theta_k} R_k(\theta)$. Then we have that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon) \leq \mathbb{P}(\bar{B}_k(\epsilon/5)) + \mathbb{P}(R_k > \delta) + \mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), R_k \leq \delta). \quad (5.17)$$

Observe that

$$\mathbb{P}(\bar{B}_k(\epsilon/5)) = (1 - G(\Theta_{\epsilon/5}))^k. \tag{5.18}$$

Observe that $N_k(B(x,\delta))$ is $Bin(k, G(B(x,\delta)))$ distributed for any $x \in \Theta$, where $Bin(k,p)$ denotes a binomial random variable with parameters k and p. Thus, given that $\theta_i = x$ for some $i \leq k$, we have that $N_k(B(x,\delta))$ is $Bin(k-1, G(B(x,\delta))) + 1$. Moreover, because $n_k = \Omega(k^q)$ and $0 < q < 1$, there exists $k_1 \in \mathbb{N}$ large enough such that $4 \leq 2n_k \leq k+3$ for all $k \geq k_1$. Thus, for $k \geq k_1$, $i \leq k$, and $x \in \Theta$, we have that

$$\begin{aligned}
\mathbb{P}(R_k(\theta_i) > \delta | \theta_i = x) &= \mathbb{P}(N_k(B(x,\delta)) \leq n_k - 1 | \theta_i = x) \\
&= \sum_{n=0}^{n_k-2} \binom{k-1}{n} (G(B(x,\delta)))^n (1 - G(B(x,\delta)))^{k-n-1} \\
&\leq n_k \binom{k-1}{n_k - 2} \alpha^{k-1}.
\end{aligned} \tag{5.19}$$

Next note that

$$\begin{aligned}
&\mathbb{P}(R_k > \delta) \\
&= \int_\Theta \mathbb{P}\left(\bigcup_{\theta \in \Theta_k} \{R_k(\theta) > \delta\} \Big| \theta_1 = x_1 \right) G(\mathrm{d}x_1) \\
&\leq \int_\Theta \mathbb{P}\left(\bigcup_{\theta \in \Theta_k \setminus \{\theta_1\}} \{R_k(\theta) > \delta\} \Big| \theta_1 = x_1 \right) G(\mathrm{d}x_1) + \int_\Theta \mathbb{P}(R_k(x_1) > \delta | \theta_1 = x_1)\, G(\mathrm{d}x_1) \\
&= \mathbb{P}\left(\bigcup_{\theta \in \Theta_k \setminus \{\theta_1\}} \{R_k(\theta) > \delta\} \right) + \int_\Theta \mathbb{P}(R_k(x_1) > \delta\} | \theta_1 = x_1)\, G(\mathrm{d}x_1).
\end{aligned}$$

Hence, proceeding recursively in a similar manner, we obtain that

$$\mathbb{P}(R_k > \delta) \leq \sum_{i=1}^{k} \int_\Theta \mathbb{P}(R_k(x_i) > \delta | \theta_i = x_i)\, G(\mathrm{d}x_i) \leq k n_k \binom{k-1}{n_k - 2} \alpha^{k-1}. \tag{5.20}$$

The second inequality follows from equation (5.19).

Suppose that $n_k \geq L k^q$ for all $k \geq 1$. Let A be a set of deterministic points in Θ and suppose that $B_k(\epsilon/5)$ and $\{R_k \leq \delta\}$ hold when $\Theta_k = A$. Then we have that

$$\begin{aligned}
&\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), R_k \leq \delta | \Theta_k = A) \\
&\quad \leq \mathbb{P}(\cup_{\theta \in \Theta_k} \{|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5\} | \Theta_k = A) \\
&\quad \leq \sum_{\theta \in A} \mathbb{P}(|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5 | \Theta_k = A).
\end{aligned} \tag{5.21}$$

The first inequality follows because the event $\{f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5)\}$ can only happen if $|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5$ for some $\theta \in \Theta_k$. Now, for each $\theta \in \Theta_k = A$, let $n_k(\theta)$ be the number of points in $B(\theta, R_k(\theta))$. Thus, for each $\theta \in A$, there exists a sequence of points $\{x_i\}_{i=1}^{n_k(\theta)}$ in Θ_k that are within a distance of $R_k(\theta)$ to θ, with $n_k(\theta)$, $R_k(\theta)$, and $x_1, \ldots, x_{n_k(\theta)}$ being deterministic given $\Theta_k = A$ (we again omit the dependency of $x_1, \ldots, x_{n_k(\theta)}$ on k and θ for notational simplicity). Thus, proceeding similarly as in the proof of equation (5.15) in Theorem 5.2, we obtain that

$$\mathbb{P}(|\hat{f}_k(\theta) - f(\theta)| > 2\epsilon/5 | \Theta_k = A) \leq \frac{C}{(Lk^q)^l}. \tag{5.22}$$

The only differences in the proof of equation (5.22) are that the second inequality follows from the fact that $R_k(\theta) \leq R_k \leq \delta$ and the final inequality uses the fact that $n_k(\theta) \geq n_k \geq Lk^q$. Combining equations (5.21) and (5.22) and recalling that $|\Theta_k| = k$ we obtain that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), R_k \leq \delta | \Theta_k = A) \leq \frac{const}{k^{ql-1}}.$$

Observe that the inequality above holds trivially when Θ_k is such that either $B_k(\epsilon/5)$ or $\{R_k \leq \delta\}$ or both do not occur. Hence, unconditioning of the expression above yields

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon, B_k(\epsilon/5), R_k \leq \delta) \leq \frac{const}{k^{ql-1}}. \tag{5.23}$$

Combining equations (5.17), (5.18), (5.20), and (5.23) shows that

$$\mathbb{P}(f(\theta_k^*) < f^* - \epsilon) \leq (1 - G(\Theta_{\epsilon/5}))^k + kn_k \binom{k-1}{n_k - 2} \alpha^{k-1} + \frac{const}{k^{ql-1}}. \tag{5.24}$$

Assumption 5.4 ensures that $\sum_{k=1}^{\infty}(1 - G(\Theta_{\epsilon/5}))^k < \infty$. Note that $n_k \leq k$ implies that $kn_k\binom{k-1}{n_k-2}\alpha^{k-1} \leq k^{n_k}\alpha^{k-1}$. Moreover, $\limsup_{k\to\infty}(k^{n_k}\alpha^{k-1})^{1/k} = \alpha < 1$ since $n_k = \Omega(k^q)$ and $q < 1$. Hence, by the root test we conclude that $\sum_{k=1}^{\infty} kn_k\binom{k-1}{n_k-2}\alpha^{k-1} < \infty$.

If $q > 1/l$, then $ql - 1 > 0$. This shows that the third term on the right-hand side of (5.24) converges to 0 as $k \to \infty$. Hence, we have shown that $f(\theta_k^*) \to f^*$ in probability as $k \to \infty$.

Furthermore, if $q > 2/l$, then $ql - 1 > 1$. This shows that the sum over k of the third term on the right-hand side of (5.24) is convergent. The first Borel-Cantelli lemma gives

that $\mathbb{P}(f(\theta_k^*) < f^* - \epsilon \text{ i.o.}) = 0$. By Theorem 4.2.2 in Chung [28] we get that $f(\theta_k^*) \to f^*$ with probability one as $k \to \infty$ since ϵ is arbitrary. ■

The comparison between Theorems 5.1 and 5.3 is very similar to that of Theorems 5.1 and 5.2 given at the end of Section 5.3, with the only difference being that structurally in the case of Theorem 5.3 we require that f is uniformly continuous and that Assumption 5.8 holds.

We now briefly compare Theorems 5.2 and 5.3. In Theorem 5.3, we do not require as strong assumptions on Θ and G as in Theorem 5.2 (recall that Assumptions 5.6 and 5.7 of Theorem 5.2 imply Assumption 5.8 of Theorem 5.3). The minimum values of l under which DSB and SSB converge (either in probability or almost surely) are the same. Finally, note that Lemma 5.2 and the proof of Theorem 5.2 show that with k large, there are with high probability $\Phi(k^q)$ sampled points within a distance r_k from each feasible solution for DSB, where q is required to be larger than $1/l$ and $2/l$ for convergence of DSB in probability and almost surely, respectively. These are the exact restrictions on l that Theorem 5.3 imposes on SSB.

The bounds on q in Theorem 5.3 restrict how slowly n_k can increase so that SSB is guaranteed to converge. This is consistent with the earlier comments that n_k should increase rapidly enough to ensure that the effects of noise are not too significant. As in Section 5.3, note that we need to control noise more tightly than bias. Moreover, as expected, a weaker condition on q is required for convergence in probability as opposed to almost sure convergence.

We now briefly discuss implementation issues regarding SSB. Suppose that all the required information to compute the estimator of the optimal solution when the search is terminated (i.e., each point's location and its objective function estimate) is stored in a list. Then a naive way of computing the estimate of the objective function value at each sampled point θ is to traverse the list and dynamically maintain a separate list that contains the distance to θ and objective function observation of the $n_k = \Omega(k^q)$ closest sampled points to θ, where k is the iteration number when the search is terminated. Hence computing the estimate of the objective function value at each sample point requires $k \log(k)$ operations

(because it takes $\log(k)$ operations to update this separate list for each sampled point in the list of all sampled points). Thus, the computation of the estimate of the optimal solution requires $O(k^2 \log(k))$ operations. Again, this can be a significant overhead if k is large. Similar research directions as for DSB can be undertaken to alleviate this problem in order to expand the applicability of the SSB method, see the discussion at the end of Section 5.3.

5.5 Numerical Examples

In this section, we compare the numerical performance of the ASR, DSB, and SSB methods to that of five other sampling-based methods due to Yakowitz and Lugosi [94] and Yakowitz, L'Ecuyer, and Vázquez-Abad [93] that also do not require much knowledge about the special structure of the objective function f (e.g., derivative information). We will see that the performance of the methods depends on the dimension of the feasible region and on the smoothness of the objective function (we say that an objective function is smooth/non-smooth if it is not difficult/difficult to identify good solutions using global search only). More specifically, in Section 5.5.1, we describe the test problems used in our numerical experiments and in Section 5.5.2, we provide implementation details for the considered approaches. Finally, in Section 5.5.3, we compare the numerical performance of the methods.

5.5.1 Test Problems

In this section, we describe our test problems. The first test problem is referred to as the *smooth* problem. It is the simulation optimization problem (1.1) with

$$f(x_1, x_2) = -\big((x_1 - 0.5)\sin(10x_1) + (x_2 + 0.5)\cos(5x_2)\big),$$

$\Theta = \{\theta = (x_1, x_2) \subset \mathbb{R}^2 : 0 \leq x_1, x_2 \leq 1\}$, and for each $\theta \in \Theta$, $h_\theta(X_\theta) = f(\theta) + X_\theta$ and X_θ is a $N(0,1)$ random variable, where $N(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. The optimal value f^* is approximately 1.502. This problem was also used by Yakowitz, L'Ecuyer, and Vázquez-Abad [93] and was included here to show that their approach is effective on "smooth" problems with low dimensional feasible regions.

The second test problem is the *two hills* problem. This is a continuous version of the

two hills problem used in Sections 3.5 and 4.5. Its objective function f is

$$f(\theta) = \max\{f_1(\theta), f_2(\theta), 0\},$$

where $f_1(\theta) = -(0.4\theta_1 - 5)^2 - 2(0.4\theta_2 - 17.2)^2 + 7$ and $f_2(\theta) = -(0.4\theta_1 - 12)^2 - (0.4\theta_2 - 4)^2 + 4$. The feasible region is given by $\Theta = \{\theta = (\theta_1, \theta_2) \in \mathbb{R}^2 : 0 \leq \theta_1, \theta_2 \leq 50\}$. The form of $h_\theta(X_\theta)$ is as for the smooth problem with X_θ being $N(0, 50)$ for all $\theta \in \Theta$. This objective function is of interest because it has two hills of different heights (4 and 7), located relatively far apart (the hill of height 4 is centered at $(30, 10)$ and the hill of height 7 is centered at $(12.5, 43)$, and separated by a flat valley (of height 0). Notice that the standard deviation of the white noise is roughly equal to the range of the objective function values. This makes the response surface highly noisy and hence this problem is relatively difficult to solve.

The third test problem is the *Rosenbrock* problem. Its objective function is given by

$$f(\theta) = -\left(\sum_{i=1}^{s-1} \left[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2\right] + 1\right).$$

The feasible region is

$$\Theta = \{\theta = (x_1, \ldots, x_s) \in \mathbb{R}^s : -10 \leq x_i \leq 10 \text{ for all } i = 1, \ldots, s\}.$$

The form of $h_\theta(X_\theta)$ is as for the other two test problems with X_θ being $N(0, 100)$ for all $\theta \in \Theta$ and $s \in \mathbb{N}^+$. This problem has a global minimum at $(1, \ldots, 1)$ and $f^* = -1$. In our numerical experiments, we use the Rosenbrock problem with $s \in \{2, 5, 10\}$ and these problems are referred to as *Rosenbrock 2D, Rosenbrock 5D,* and *Rosenbrock 10D*, respectively.

5.5.2 Algorithm Implementation

In this section we provide implementation details for the ASR, DSB, and SSB approaches and for the methods we compare them with, namely the YL method and two versions of each of the convergent and heuristic methods of Yakowitz, L'Ecuyer, and Vázquez-Abad [93].

We first describe the implementation details related to ASR. The sampling procedure we considered is as follows. In iteration $k = M(i)$, with probability $g > 0$, a new solution θ is

sampled uniformly from Θ, and with probability $1-g$, a new solution is sampled uniformly from $N(\theta^*_{k-1})$, where $N(\theta) = N((x_1,\ldots,x_s)) = \{(x'_1,\ldots,x'_s) \in \Theta : |x_i - x'_i| \leq r \text{ for all } i = 1,\ldots,s\}$ for all $\theta \in \Theta$ (the first point is sampled uniformly from Θ). The newly sampled point θ is accepted if $f_K(\theta) \geq \hat{f}_{k-1}(\theta^*_{k-1}) - \delta$, where $K \in \mathbb{N}^+$ and $\delta > 0$ (we assume that the first sampled point is always accepted). The resampling procedure in iteration k is as follows. Let $k' = M(m_k)$, i.e., the last iteration number prior to iteration k when a new point is sampled. Then, a point $\theta \in \Theta_k$ is resampled in iteration k with probability

$$p_k(\theta) = \frac{\exp(\hat{f}_{k'}(\theta)/T(k'))}{\sum_{\theta' \in \Theta_k} \exp(\hat{f}_{k'}(\theta')/T(k'))},$$

where $T(k') = T/\log(k'+1)$ with $T > 0$. This resampling procedure is also used by Yakowitz and Lugosi [94]. Observe that this procedure puts more weight on the points that have better estimated objective function values, and that as the simulation effort goes to infinity, only points with the highest estimated objective function values are sampled. Moreover, the resampling procedure is only updated when a new point is sampled. Finally, for each $i \in \mathbb{N}^+$, we let $K(i) = \lceil Ck^c \rceil$, where $c, C > 0$ and $\lceil \cdot \rceil$ denotes a ceiling function.

We next describe the implementation details for the DSB, SSB, and YL methods. For DSB and SSB, the metric d is Euclidean and G is the uniform distribution on Θ. For the DSB method, $r_k = Ck^{-p/s}$ for all $k \in \mathbb{N}^+$, while for the SSB method, $n_k = \lceil Ck^q \rceil$ for all $k \in \mathbb{N}^+$, where $C > 0$. In the YL method, the density $p(x)$ is uniform on Θ and $T(n)$ is given above. This implementation is also used by Yakowitz and Lugosi [94].

We now briefly describe the implementation details for the methods of Yakowitz, L'Ecuyer, and Vázquez-Abad [93]. For their methods, the authors suggest collecting m^*_N points within the feasible region, where N is the simulation budget, and then expend this simulation budget on identifying the best point within the chosen collection of points. In their work, the collection of points used on problems with feasible regions that are generalized hypercubes in two or higher dimensions is the quasi-random set

$$\bar{\Theta} = \left\{ (x_1,\ldots,x_s) \in \mathbb{R}^s \middle| x_j \in \left\{ l_j + \frac{(u_j - l_j)}{2k}, l_j + \frac{3(u_j - l_j)}{2k}, \ldots, l_j + \frac{(2k-1)(u_j - l_j)}{2k} \right\} \right\},$$

where $k = \lfloor (m^*_N)^{1/s} \rfloor$ and l_j and u_j are the lower and upper bounds on x_j, respectively, for $j = 1,\ldots,s$. Note that $|\bar{\Theta}| \leq m^*_N$, and hence we may collect a smaller number of points

than suggested by their approach. This implementation is also used in Yakowitz, L'Ecuyer, and Vázquez-Abad [93]. We also implement their methods with the set of sampled points being a set of m_N^* independent and uniformly distributed points over Θ. We consider this implementation because it is of interest to understand the effects of quasi-random collections of points on the performance of these methods, and also this provides a fairer comparison to the other approaches (because it eliminates the element of "luck" with respect to whether the collection of quasi-random points is such that it contains points in good areas). Subsequently, we refer to these methods as either being Q or R depending on how the collection of points is generated. The methods of Yakowitz, L'Ecuyer, and Vázquez-Abad [93] are either adaptive (if simulation effort is expended adaptively; these methods are not guaranteed to converge) or nonadaptive (if each sampled point receives the same amount of simulation effort; these methods are convergent). Thus, we subsequently refer to these methods as *Heuristic* and *Convergent* depending on their convergence guarantee. Overall we consider four versions of their methods (i.e., all possible combinations of Q and R with Heuristic and Convergent).

An effort was made to select good parameter values for each algorithm. In particular, the parameter values for the ASR method were optimized on the smooth problem and were used on all the other problems except that the value of r (the radius of the "local" neighborhood) was adjusted for the size of the underlying feasible region (i.e., it is the value of r used for the smooth problem multiplied by $(u_1 - l_1)$). More specifically, the parameter values for ASR are $b = 1.1$, $c = 0.5$, $C = 1$, $g = 0.5$, $\delta = 0.01$, $K = 10$, and $T = 0.1$, with r being 0.02 for the smooth problem, 1.0 for the two hills problem, and 0.4 for all the Rosenbrock problems. The parameter values for the DSB, SSB, and YL methods were optimized for each particular problem over a set of substantially different values. The resulting parameter values are given in Table 5.1 (Ros stands for Rosenbrock and the definitions of the parameters b, p, and C' of the YL method can be found in Yakowitz and Lugosi [94]). Note that less effort has been put into optimizing the performance of ASR when compared to the DSB, SSB, and YL methods, which suggests that the performance ASR may be robust. Finally, the parameter values for the Q Heuristic, Q Convergent, R Heuristic, and R Convergent methods were

chosen as suggested by Yakowitz, L'Ecuyer, and Vázquez-Abad [93]. In particular, the parameter m_N^* is $\lceil 10(N/\log(N))^{s/(s+4)} \rceil$, where s is the dimension of the problem, with the exception that it is $\lceil 10N^{2/5} \rceil$ for the smooth, two hills, and Rosenbrock 2D problems (i.e., all problems with $s = 2$) for the Q and R Heuristic methods. These parameter values satisfy the conditions in the convergence results for the ASR, DSB, SSB, YL, and Q and R Convergent methods. The reason why we chose a different m_N^* for all the two dimensional problems for the Q and R Heuristic methods is that Yakowitz, L'Ecuyer, and Vázquez-Abad [93] also considered this sequence and obtained good results for the Q Heuristic method on the smooth problem. On the other hand, they did not specify how to pick such a sequence for the Q Heuristic method for an arbitrary dimension s, and hence for the Rosenbrock 5D and 10D problems we use the same m_N^* for both the Heuristic and Convergent methods.

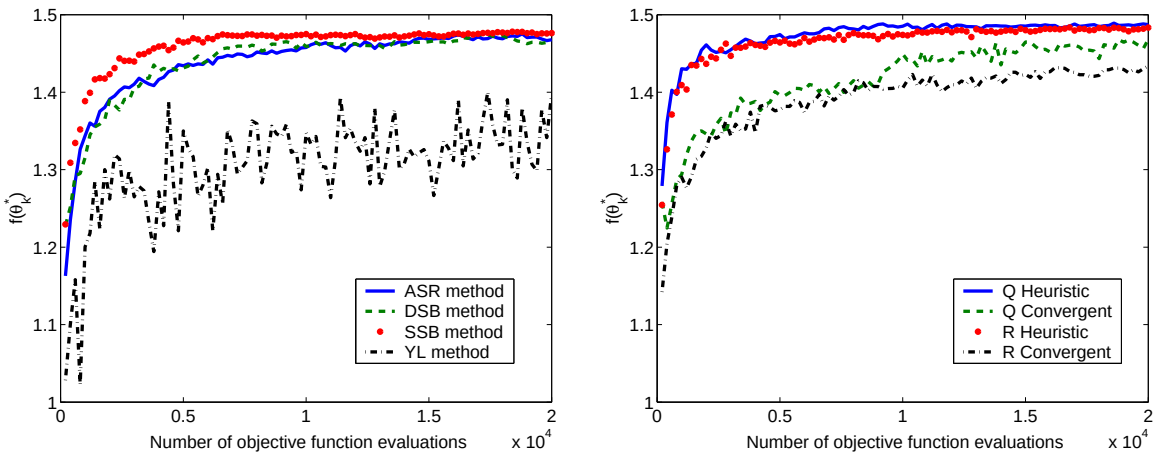
Table 5.1: Parameter values for the DSB, SSB, and YL methods

Approach	Parameters	Smooth	Two Hills	Ros 2D	Ros 5D	Ros 10D
DSB	p	0.5	0.75	0.25	0.25	0.75
	C	0.822	115.866	1.507	5.589	24.159
SSB	q	0.5	0.25	0.75	0.75	0.75
	C	2.121	16.870	0.018	0.090	19.191
YL method	b	1.5	1.5	1.5	1.5	1.5
	p	5	5	5	5	5
	C'	0.01	0.01	0.01	0.01	0.01
	T	0.1	1	10	10	10

The performance of the algorithms is averaged over 100 independent replications for all the problems. Their performance is documented by plotting 100 pairs (x, y), where $x \in \{0.01N, 0.02N, \ldots, N\}$, N is the simulation budget, and y is the average objective function value at the estimate of the optimal solution after x objective function observations have been collected. Also, note that the performance of the Q and R Convergent (Heuristic) methods has been optimized for each particular value of x because these methods require the knowledge of the overall simulation budget N (since the number of sampled points m_N^* depends on it). This favors these four approaches over the other methods because the performance of the other methods is not optimized for each particular run length.

5.5.3 Algorithm Comparison

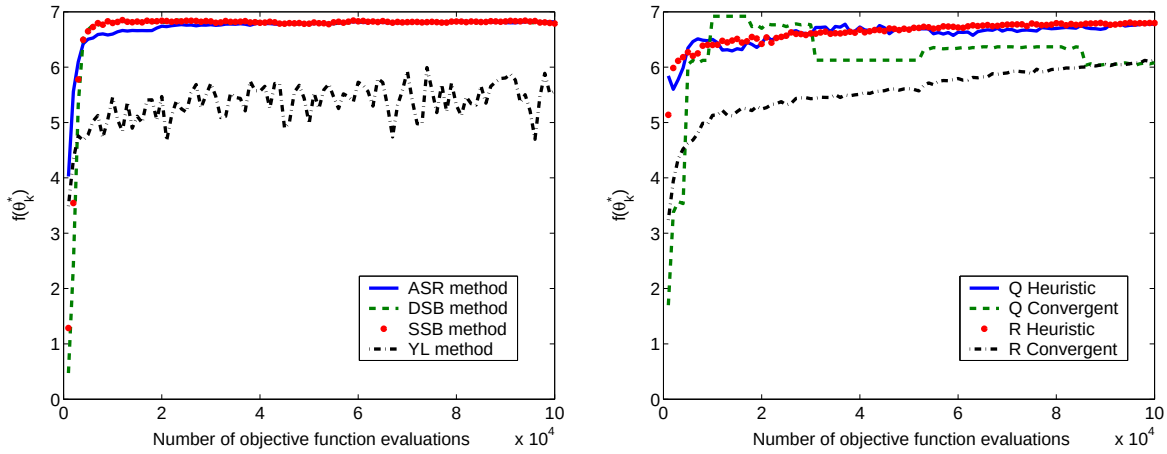
In this section we compare the numerical performance of ASR, DSB, and SSB to that of the YL method and the Q and R Heuristic and Convergent methods. Figures 5.1 through 5.5 show the empirical performance of these methods on the smooth, two hills, Rosenbrock 2D, Rosenbrock 5D, and Rosenbrock 10D problems, respectively. Observe that for the three Rosenbrock problems, we plot $-f(\theta_k^*)$ on a logarithmic scale (rather than $f(\theta_k^*)$ on a linear scale) as the simulation effort increases to facilitate comparisons. Thus, smaller values are better for Figures 5.3 through 5.5, while larger values are better for Figures 5.1 and 5.2. Finally, the sequence in which the numerical results are presented moves from problems with more "smooth" f and low dimensions to problems with less "smooth" f and higher dimensions. These examples illustrate that ASR can be very effective, especially when f is "non-smooth" (in the sense that sampling a solution in a "good" subregion using global search is unlikely).



(a) ASR, DSB, SSB, and YL methods (b) Q and R Heuristic and Convergent methods

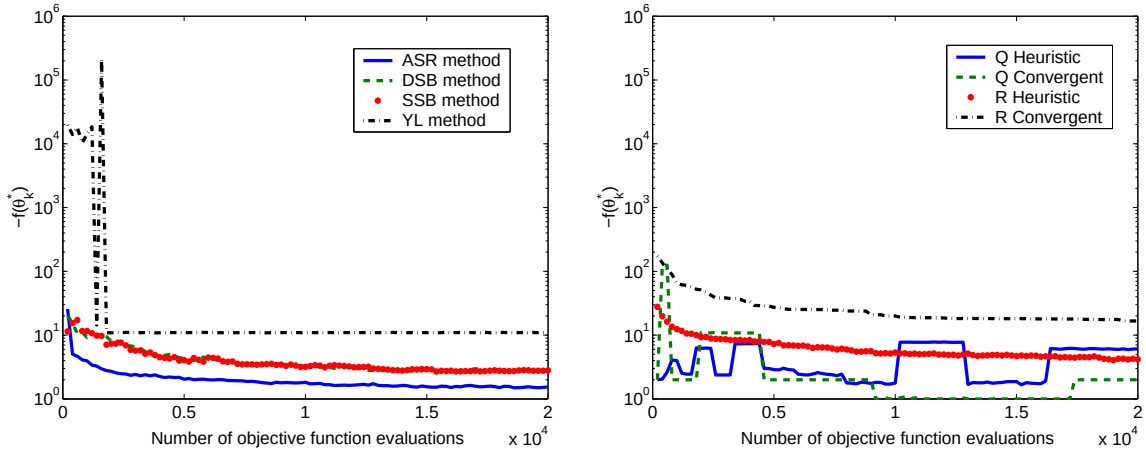
Figure 5.1: Performance of the optimization methods on the smooth problem

It is clear from Figures 5.1 through 5.5 that the ASR method has a comparatively good performance, especially on the higher dimensional problems with less "smooth" objective functions. Moreover, the ASR method has relatively low overhead compared to the other approaches. Thus, we believe that ASR is overall the most effective approach among the



(a) ASR, DSB, SSB, and YL methods (b) Q and R Heuristic and Convergent methods

Figure 5.2: Performance of the optimization methods on the two hills problem

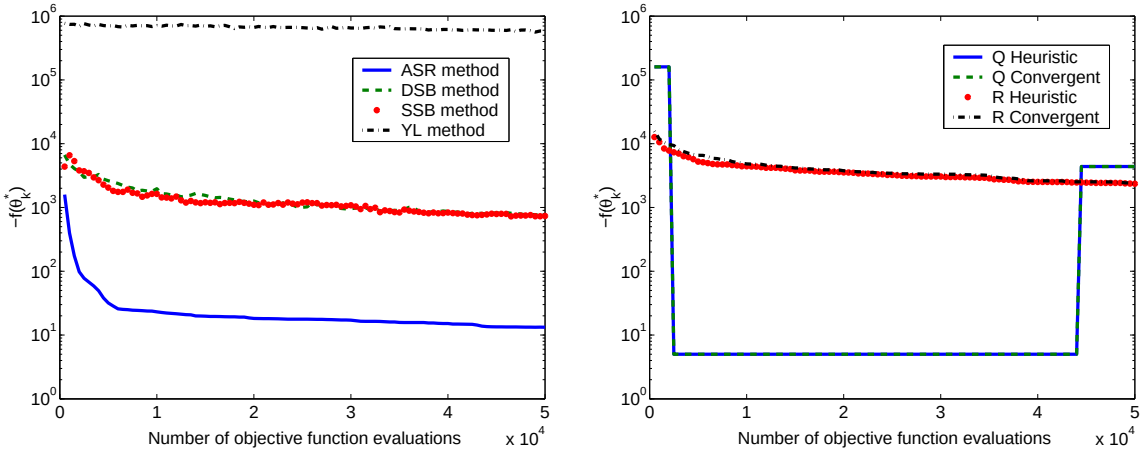


(a) ASR, DSB, SSB, and YL methods (b) Q and R Heuristic and Convergent methods

Figure 5.3: Performance of the optimization methods on the Rosenbrock 2D problem

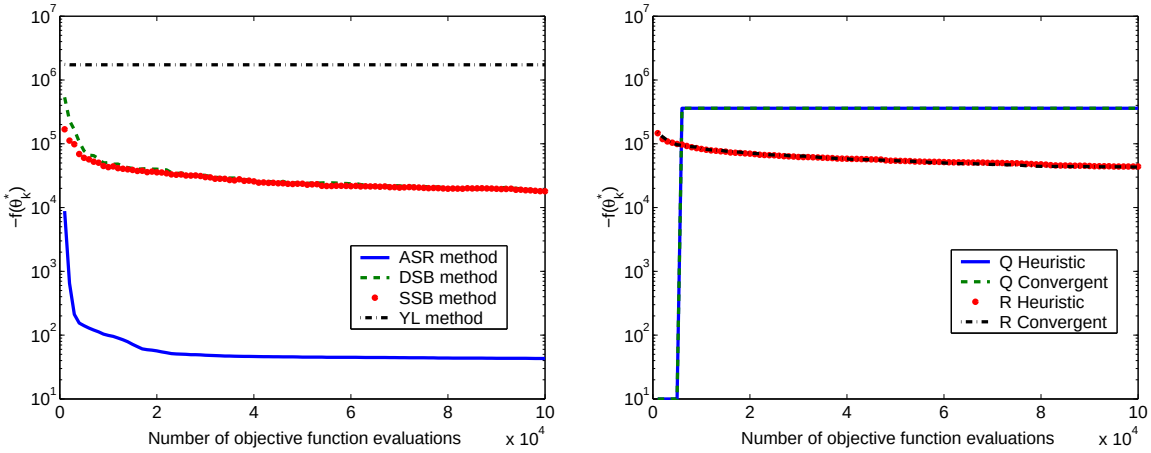
methods considered in this numerical study. We believe the main reason for the effective performance of ASR relative to the other methods is that it can incorporate an adaptive local search component into its sampling strategy, which may allow for a more efficient search of the feasible space (the other methods lack this feature).

Figures 5.1 through 5.5 also show that the SSB method has similar or better performance than the DSB method (this is consistent with the discussion in Section 5.4), with SSB performing significantly better on the smooth problem. Hence, if obtaining an additional



(a) ASR, DSB, SSB, and YL methods (b) Q and R Heuristic and Convergent methods

Figure 5.4: Performance of the optimization methods on the Rosenbrock 5D problem



(a) ASR, DSB, SSB, and YL methods (b) Q and R Heuristic and Convergent methods

Figure 5.5: Performance of the optimization methods on the Rosenbrock 10D problem

observation of the objective function is relatively expensive compared to the overhead of the chosen optimization method (as is typically the case in simulation optimization), then SSB is overall preferable to DSB (in our implementation it is more expensive to compute the estimator of the optimal solution for SSB than for DSB) .

We now discuss the performance of the five approaches we compare the ASR, DSB, and SSB methods with. Figures 5.1 through 5.5 show that the YL method usually behaves the worst of all eight methods, largely due to a poor choice of the estimator of the optimal

solution. Also the average performance of the R Heuristic and Convergent methods is relatively smooth on all the problems, but this is only true for the Q Convergent method on the smooth problem and for the Q Heuristic method on the smooth and two hills problems. On the other problems, the performance of the Q Heuristic and Convergent methods is both "jumpy" and non-monotonic. In other words, Figures 5.2 through 5.5 show that the performance of the Q Heuristic and Convergent methods does not necessarily improve as the simulation effort grows, and, indeed, can get significantly worse. Both the jumps and the non-monotonicity in the performance of the Q Heuristic and Convergent methods happen due to the nature of the generation of the quasi-random points (see Section 5.5.2 for more details). In particular, depending on how many points the methods generate, one can be either lucky or unlucky with respect to whether the quasi-random set of points contains points in good areas. The effects of this element of luck are more pronounced in problems that are either not smooth or high dimensional because identifying "good" points is more difficult in such settings, and do not average out (because the point set is deterministic and is consequently used in all replications of the Q Heuristic and Convergent methods).

Note also that when the Heuristic and Convergent methods collect different numbers of points (see Figures 5.1 through 5.3), the Heuristic methods usually perform better than their Convergent counterparts (except for the Q Heuristic and Convergent methods on the Rosenbrock 2D problem). Also, Figures 5.4 and 5.5 suggest that when the Heuristic and Convergent methods collect the same number of points their performance is similar. However, this is not true in general (see, for instance, Tables 2 and 3 in Yakowitz, L'Ecuyer, and Vázquez-Abad [93]). In general, the Q methods behave better on the "smooth" problems with low dimensional feasible regions than their R counterparts. On the other hand, on the "non-smooth" problems with high dimensional feasible regions, the difference in the performance of the Q and R methods can go either way depending on the "luck" in generating the set of points being compared. Finally, note that the performance of the Q Convergent method on the smooth problem reported in Figure 5.1 is better than the performance of the same approach on this problem documented in Yakowitz, L'Ecuyer, and Vázquez-Abad [93].

Recall that besides the ASR, DSB, and SSB methods considered in this chapter, the Q and R Convergent methods and the YL method are the only other provably convergent methods. We now compare the ASR, DSB, and SSB approaches with these three methods. It is clear from Figures 5.1 through 5.5 that ASR, DSB, and SSB perform better than the YL and R Convergent methods, while the comparison with the Q Convergent method depends on the luck associated with the set of points sampled by the Q Convergent method. In particular, the ASR, DSB, and SSB methods dominate the Q convergent method on the smooth problem and perform better for the most simulation levels on the two hills and Rosenbrock 10D problems. We conclude this section by pointing out that the main reason why the ASR, DSB, and SSB approaches become comparatively better than the Q and R Heuristic and Convergent methods as we move through the numerical results (and the problems become more difficult) is that ASR has an adaptive local search component, and DSB and SSB sample more points than the Q and R Heuristic and Convergent methods. Consequently, it is easier for our approaches to identify good subregions of the feasible space.

5.6 Conclusions

In this chapter, we presented three random search methods for continuous simulation optimization. The effects of noise in our approaches are either reduced by occasional resampling of already sampled solutions or by averaging observations in balls that shrink with time. Our ASR method is adaptive and its sampling strategy may involve local search, while the DSB and SSB approaches are based on pure random search, with the only difference between them being the estimator of the optimal solution. We proved that all three methods are convergent, both in probability and almost surely. Finally, we demonstrated the effectiveness of our approaches (especially the ASR approach) when compared to some other methods available in the literature. Our approaches performed especially well on difficult problems for which sampling a solution in a "good" subregion using global search is unlikely.

CHAPTER VI

CONTRIBUTIONS AND FURTHER RESEARCH

This thesis is concerned with adaptive random search methods for simulation optimization. The methods are adaptive in that they use all the information gathered so far to decide how to expend simulation effort in the current iteration. By contrast, most of the earlier methods devised for solving simulation optimization problems are Markovian in that the algorithmic decisions in the current iteration can depend only on the objective function observations collected in the current iteration. One of the main reasons for this is that it is easier to show the convergence of such (Markovian) methods (by utilizing the available machinery for analyzing Markov chains). In this dissertation, we not only develop adaptive and convergent random search methods, but we also show that this adaptivity can be useful from an empirical perspective. The outline of this chapter is as follows. In Section 6.1, we briefly summarize the contributions of this thesis, while in Section 6.2 we describe some possible future research directions for our work.

6.1 Contributions

In Chapter 3, we discussed desirable features that a simulation optimization algorithm should possess to have good empirical performance. In particular, our approach to solving simulation optimization problems emphasizes maintaining an appropriate balance between exploration, exploitation, and estimation. We also developed two new almost surely convergent random search methods possessing the desired features. These methods are intuitive, simple, flexible enough to allow an end-user to exploit the structure inherent in the optimization problem at hand, and also exhibit attractive empirical performance.

In Chapter 4, we presented a general framework based on averaging for designing adaptive and almost surely convergent random search methods for discrete simulation optimization. The objective function estimate at any solution is the average of all observations collected at this solution so far. We also developed two new variants of the simulated

annealing (SA) algorithm and discussed their convergence. These analyses provided increased theoretical understanding of SA with decreasing cooling schedule for deterministic and stochastic optimization. Moreover, via numerical examples involving the proposed SA algorithms, we showed that averaging together with adaptiveness in expending simulation effort can be effective.

In Chapter 5, we presented three random search methods for continuous simulation optimization. The adaptive search with resampling (ASR) method is adaptive and its sampling strategy may involve local search, while the deterministic shrinking ball (DSB) and stochastic shrinking ball (SSB) approaches are based on pure random search, with the only difference between them being the estimator of the optimal solution (the DSB method was originally proposed and analyzed by Baumert and Smith [20]). We also presented conditions under which all three methods are convergent, both in probability and almost surely. Finally, we demonstrated the empirical effectiveness of the approaches when compared to some other random search methods available in the literature. The ASR approach in particular performed especially well on difficult problems for which sampling a solution in a good subregion of the feasible space using global search is unlikely.

6.2 Future Research

The following research directions could be undertaken with regard to improving the numerical performance of our methods presented in Chapter 3:

1. Develop better ways for adaptively selecting the number of observations to be collected by the R-BEESE and A-BEESE methods at sampled solutions, taking into account the observed quality and variability of the objective function estimates at all sampled points.

2. Numerically test different local search schemes. In particular, it might be desirable to adaptively control the size of the local neighborhood depending on the cardinality of the feasible region and the amount of the simulation effort already expended. For instance, it would be interesting to document the effects of the local neighborhood structure proposed by Hong and Nelson [51] on the numerical performance of our

approaches.

With regard to our work in Chapter 4, we are interested in investigating the effects of averaging and adaptivity in the context of other simulation optimization approaches (beyond SA). Finally, the following directions could be undertaken to further improve the applicability and theoretical and practical understanding of the three approaches for continuous simulation optimization that were discussed in Chapter 5:

1. Investigate resampling procedures for the ASR method that can take better advantage of the available information (such as the estimated variance of the observed objective function values at each sampled point) with the goal of improving the empirical performance of the method.

2. Devise more efficient ways of calculating (or approximating) the estimator of the optimal solution for the DSB and SSB methods.

3. Extend the DSB and SSB algorithms to be more adaptive to the available information, while preserving their convergence guarantees.

4. Prove the convergence of the ASR, DSB, and SSB methods when applied to solve optimization problems involving steady-state simulations. One possible approach to address this issue is to consider batching to achieve approximate independence of the individual objective function observations at each solution. It is also of interest to understand the impact of this approximation on the numerical performance of the proposed approaches.

5. Expand numerical study to compare the proposed approaches with other optimization methods available in the literature and on other (simulation) optimization problems.

6. Investigate whether it is more efficient to solve continuous simulation optimization problems directly or by first discretizing the feasible region and then solving the resulting discrete optimization problem using an optimization method designed for discrete simulation optimization.

APPENDIX A

PROOFS OF LEMMAS 4.1 THROUGH 4.3

Proof of Lemma 4.1: Suppose that $L = 0$. Fix $\theta \in \Theta_L$ and $\theta' \in N(\theta)$ such that $f(\theta) > f(\theta')$ (this is possible due to Assumption 4.14). From Assumption 4.11 we have that the graph G is connected. Therefore, there exists a finite sequence $\{\theta_i\}_{i=0}^k$ in Θ such that $\theta_0 = \theta_k = \theta$, $\theta_1 = \theta'$, and $\theta_i \in N(\theta_{i-1})$ for $i = 1, \ldots, k$. Because $L = 0$, we also have that $\hat{f}(\theta_0) \leq \hat{f}(\theta_1) \leq \cdots \leq \hat{f}(\theta_k)$ with probability one. This implies that $\hat{f}(\theta) = \hat{f}(\theta')$ almost surely and thus, $f(\theta) = f(\theta')$. This provides a contradiction and the proof is complete. ■

Proof of Lemma 4.2: The proof of this lemma consists of three steps. We first find a bound on the one-step transition probability from any $\theta \in \Theta$ to any $\theta' \in N(\theta)$ for a large enough transition number. Assumptions 4.2 and 4.11 imply that there exists $n_0 \in \mathbb{N}$ such that if $n \geq (n_0 - 1)r$, then $Q_n(\theta, \theta') \geq q$ for all $\theta \in \Theta$ and $\theta' \in N(\theta)$. Thus, if $\theta' \in N(\theta)$ and $n \geq (n_0 - 1)r$, then from (4.5) we have that

$$\mathbf{P}_n(\theta, \theta') \geq Q_n(\theta, \theta') \exp(-\mathbb{E}[\hat{f}(\theta) - \hat{f}(\theta')]^+ / T_n) \geq q \exp(-L/T_n). \tag{A.1}$$

The first inequality follows from Jensen's inequality and the second one follows from (4.3) and the choice of n.

Secondly, we compute a bound on the probability that Algorithm 4.3 stays at the current iterate provided that it is not in the set Θ_L. Fix $\theta \in \Theta \setminus \Theta_L$. Since $\theta \notin \Theta_L$, there exists $\theta' \in N(\theta)$ such that $f(\theta) > f(\theta')$. Let $A(\theta, \theta') = \{\omega \in \Omega : \hat{f}(\theta) > \hat{f}(\theta')\}$ and observe that $\mathbb{P}(A(\theta, \theta')) > 0$. Then we have

$$\mathbb{E}[\exp(-[\hat{f}(\theta) - \hat{f}(\theta')]^+ / T_n)] = \mathbb{P}(A(\theta, \theta')^c) + \int_{A(\theta,\theta')} \exp\left(-\frac{\hat{f}(\theta) - \hat{f}(\theta')}{T_n}\right) \mathrm{d}\mathbb{P}.$$

Assumption 4.12 and the monotone convergence theorem give that

$$\lim_{n\to\infty} \mathbb{E}\left[\exp(-[\hat{f}(\theta) - \hat{f}(\theta')]^+ / T_n)\right] = \mathbb{P}(A^c(\theta, \theta')) < 1. \tag{A.2}$$

For $n \geq (n_0 - 1)r$, we have

$$\begin{aligned}
\mathbf{P}_n(\theta, \theta) &= \sum_{\gamma \in N_n(\theta)} Q_n(\theta, \gamma)(1 - \mathbb{E}[\exp(-[\hat{f}(\theta) - \hat{f}(\gamma)]^+ / T_n)]) \\
&\geq Q_n(\theta, \theta')(1 - \mathbb{E}[\exp(-[\hat{f}(\theta) - \hat{f}(\theta')]^+ / T_n)]) \\
&\geq q(1 - \mathbb{E}[\exp(-[\hat{f}(\theta) - \hat{f}(\theta')]^+ / T_n)]). \qquad \text{(A.3)}
\end{aligned}$$

The first equality follows from (4.5) and the last inequality holds by the choice of n.

By (A.2) and (A.3), there exists $n_0(\theta) \geq n_0$ such that $n \geq (n_0(\theta) - 1)r$ implies that $\mathbf{P}_n(\theta, \theta) \geq q\mathbb{P}(A(\theta, \theta'))/2$. Lemma 4.1 and Assumption 4.12 ensure that there exists $n_1(\theta) \in \mathbb{N}$ such that $\exp(-L/T_n) \leq \mathbb{P}(A(\theta, \theta'))/2$ provided that $n \geq (n_1(\theta) - 1)r$. Let $n(\theta) = \max\{n_0(\theta), n_1(\theta)\}$. Therefore, if $n \geq (n(\theta) - 1)r$, then

$$\mathbf{P}_n(\theta, \theta) \geq q \exp(-L/T_n). \qquad \text{(A.4)}$$

Let $n_1 = \max_{\theta \in \Theta \setminus \Theta_L} n(\theta)$. Assumption 4.2 insures that $n_1 \in \mathbb{N}$ and hence we have that equation (A.4) holds for all $n \geq (n_1 - 1)r$ and $\theta \in \Theta \setminus \Theta_L$.

Finally, observe that equations (A.1) and (A.4), Assumption 4.14, and the definition of r ensure that starting in iteration $n \geq (n_1 - 1)r$, the Markov chain W can reach every point $\theta' \in \Theta$ from any point $\theta \in \Theta$ in exactly r transitions, where each transition occurs from a point $x \in \Theta$ to a point $y \in N(x) \cup \{x\}$. Thus, if $n \geq n_1 r$, then

$$\mathbf{P}_{n-r,n}(\theta, \theta') \geq \prod_{i=n-r}^{n-1} (q \exp(-L/T_i)) \geq q^r \exp(-rL/T_{n-1}).$$

The first inequality above follows from equations (A.1) and (A.4), while the second one follows from Assumption 4.12. ■

Proof of Lemma 4.3: It is obvious from Assumption 4.12 that (i) implies (ii). To prove the converse, fix $k \in \mathbb{N}$. Then $k = ra + b$ where $a \in \mathbb{N}$ and $0 \leq b < r$. Thus,

$$\begin{aligned}
r \sum_{n=1}^{\infty} \exp\left(-\frac{rL}{T_{nr+k-1}}\right) &= r \sum_{n=a+1}^{\infty} \exp\left(-\frac{rL}{T_{nr+b-1}}\right) \geq r \sum_{n=a+1}^{\infty} \exp\left(-\frac{rL}{T_{(n+1)r}}\right) \\
&\geq \sum_{n=r(a+2)}^{\infty} \exp\left(-\frac{rL}{T_n}\right) = +\infty.
\end{aligned}$$

Both inequalities follow from Assumption 4.12, and the last equality follows from (ii).

■

APPENDIX B

PROOF OF THEOREM 4.5 AND EXTENSION OF ASSUMPTION 4.16

Proof of Theorem 4.5: Note that without loss of generality, we can assume that Assumption 4.14 holds and hence that $\Theta \neq \Theta^*$. By Assumption 4.1, we can assume that $\Omega = \Omega_d \times \Omega_s$, where $\Omega_d = \prod_{n=0}^{\infty} \Omega_n$. Let $\mathbb{P}_d$ and $\mathbb{P}_s$ be the probability measures on Ω_d and Ω_s, respectively.

Next we briefly outline the proof. Let $\tilde{\Omega}_s$ and $\bar{\Omega}$ be as defined in the proof of Theorem 4.1. Observe that Assumptions 4.2 and 4.4 imply that $\mathbb{P}(\tilde{\Omega}_s) = 1$. Also, Assumptions 4.2 and 4.5 ensure that $\mathbb{P}(\bar{\Omega}) = 1$. First, for each $\omega_s \in \tilde{\Omega}_s$, we will construct a subset $\tilde{\Omega}_d(\omega_s)$ of Ω_d that possesses desired properties and satisfies $\mathbb{P}_d(\tilde{\Omega}_d(\omega_s)) = 1$. Second, we will show that Algorithm 4.4 converges to the set Θ^* for every $\omega \in (\tilde{\Omega}_d(\omega_s) \times \{\omega_s\}) \cap \bar{\Omega}$. Finally, we verify that $\mathbb{P}\{\cup_{\omega_s \in \tilde{\Omega}_s} \tilde{\Omega}_d(\omega_s) \times \{\omega_s\}\} = 1$.

For each $\theta \in \Theta \setminus \Theta_L$, let γ_θ be some solution in $N(\theta)$ such that $f(\theta) > f(\gamma_\theta)$. Define $\alpha = \min_{\theta \in \Theta \setminus \Theta_L} \{f(\theta) - f(\gamma_\theta)\}$ and let $\epsilon = \min\{\frac{1}{2}(L' - L), \frac{1}{3}\alpha\}$. Assumption 4.14 and the definitions of L' and L ensure that $\epsilon > 0$.

Note that without loss of generality, we can assume that Θ is of the form $\{1, 2, \ldots, b\}$, where $b = |\Theta|$. For each $k \in \mathbb{N}$ and $\theta \in \Theta$, let $\tilde{f}_k(\theta) = \sum_{i=1}^{k} h_\theta(X_\theta^i)/k$. Moreover, for all $\omega_s \in \Omega_s$, $x \in \Theta$, and $k, k_1, \ldots, k_b \in \mathbb{N}$, let $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$ denote Algorithm 4.4 with the objective function observations collected under ω_s initialized (in Step 0 of Algorithm 4.4) with the starting point x, iteration number $n = k$, $C_n(i) = k_i$, and $\Sigma_n(i) = k_i \times \tilde{f}_{k_i}(i, \omega_s)$ for all $i \in \{1, 2, \ldots, b\}$. Note that in any iteration $n \geq k$, the first $C_n(i)$ objective function observations at each feasible point $i \in \Theta$ are available, and consequently can be used in the execution of algorithmic decisions. Assumption 4.2 implies that for each $\omega_s \in \tilde{\Omega}_s$ there exists $k(\omega_s) \in \mathbb{N}$ such that $|\tilde{f}_{k'}(\theta, \omega_s) - f(\theta)| < \epsilon$ for all $\theta \in \Theta$ and $k' \geq k(\omega_s)$.

In the next few lemmas, we will verify that $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$ samples each feasible solution infinitely often with probability one for all $\omega_s \in \tilde{\Omega}_s$, $x \in \Theta$, and $k \in \mathbb{N}$, provided

that $k_i \geq k(\omega_s)$ for $i = 1, \ldots, b$ (so that each feasible point has a relatively precise estimate of the objective function value at it). Fix $\omega_s \in \tilde{\Omega}_s$, $x \in \Theta$, and $k, k_1, \ldots, k_b \in \mathbb{N}$, and let $\{\bar{\theta}_n\}_{n=k}^{\infty}$, $\{\bar{\theta}'_n\}_{n=k}^{\infty}$, $\{\bar{C}_n(\theta)\}_{n=k}^{\infty}$, and $\{\bar{f}_n(\theta)\}_{n=k}^{\infty}$, for each $\theta \in \Theta$, be the stochastic processes generated by $Alg(\omega_s, x, k, \{k_i\}_{i=1}^{b})$ corresponding to the stochastic processes $\{\theta_n\}$, $\{\theta'_n\}$, $\{C_n(\theta)\}$, and $\{\hat{f}_n(\theta)\}$, for each $\theta \in \Theta$, generated by Algorithm 4.4. Although the objective function observations are fixed (because ω_s is fixed), these processes are still random because of the randomness inherent in the underlying algorithm. Consequently, these processes are defined on the sample space Ω_d with associated probability measure $\mathbb{P}_d$ (note that this relies on Assumption 4.1, and consequently Assumption 4.1 is used implicitly in the proofs of Lemmas B.1 through B.4). Thus, the probability measures referred to in Lemmas B.1 through B.4 are, indeed, $\mathbb{P}_d$. To simplify notation, this dependence will be suppressed. Also, for all $m, n, k'_1, \ldots, k'_b \in \mathbb{N}$, $\theta \in \Theta$, and $\{x_l\}_{l=k}^{n-1} \subset \Theta$ such that $n \geq k$ and $k \leq m \leq n$, define an event

$$A(m, n, \theta, \{k'_i\}_{i=1}^{b}) = \{\bar{\theta}_n = \theta, \bar{\theta}_l = x_l \text{ for } l = k, \ldots, n-1, \bar{C}_m(i) = k'_i \text{ for } i = 1, \ldots, b\}.$$

We first derive a bound on the one-step transition probability from the current point θ to the candidate point $\theta' \in N(\theta)$ for $Alg(\omega_s, x, k, \{k_i\}_{i=1}^{b})$ for a large enough transition number.

Lemma B.1. *Suppose that Assumptions 4.1, 4.2, and 4.11 hold. Then, for all $\theta \in \Theta$ and $\theta' \in N(\theta)$, there exists $n_0 \in \mathbb{N}$ with $(n_0 - 1)r \geq k$ such that for $n \geq (n_0 - 1)r$, we have that*

$$\mathbb{P}(\bar{\theta}_{n+1} = \theta' | A(m, n, \theta, \{k'_i\}_{i=1}^{b})) \geq q \exp(-L'/T_n),$$

provided that $k \leq m \leq n$, $k'_i \geq k(\omega_s)$ for $i = 1, \ldots, b$, and $\mathbb{P}(A(m, n, \theta, \{k'_i\}_{i=1}^{b})) > 0$.

Proof: Assumptions 4.2 and 4.11 imply that there exists $n_0 \in \mathbb{N}$ such that if $n \geq (n_0 - 1)r$, then $Q_n(\theta, \theta') \geq q$ for all $\theta \in \Theta$ and $\theta' \in N(\theta)$. Now note that if $n \geq k$, then for all $\omega_d \in \Omega_d$,

we have that

$$\begin{aligned}
[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\theta')]^+ &= [\tilde{f}_{\bar{C}_{n+1}(\theta)}(\theta, \omega_s) - \tilde{f}_{\bar{C}_{n+1}(\theta')}(\theta', \omega_s)]^+ \\
&\leq [f(\theta) + \epsilon - f(\theta') + \epsilon]^+ \\
&\leq 2\epsilon + [f(\theta) - f(\theta')]^+ \\
&\leq 2\epsilon + L \\
&\leq L'. \qquad \text{(B.1)}
\end{aligned}$$

The first inequality follows from the fact that $\bar{C}_{n+1}(\theta), \bar{C}_{n+1}(\theta') \geq k(\omega_s)$. The third inequality follows from the definition of L, while the last inequality holds by the choice of ϵ.

Now let $\theta \in \Theta$, $\theta' \in N(\theta)$, $n \geq (n_0 - 1)r$, and $n \geq k$. We have that

$$\begin{aligned}
&\mathbb{P}(\bar{\theta}_{n+1} = \theta' | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}(\bar{\theta}_n' = \theta', U_n \leq \exp(-[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\theta')]^+ / T_n) | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}(U_n \leq \exp(-[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\theta')]^+ / T_n) | A(m, n, \theta, \{k_i'\}_{i=1}^b), \bar{\theta}_n' = \theta') \\
&\quad \times \mathbb{P}(\bar{\theta}_n' = \theta' | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \\
&\geq \mathbb{P}(U_n \leq \exp(-L'/T_n) | A(m, n, \theta, \{k_i'\}_{i=1}^b), \bar{\theta}_n' = \theta') \mathbb{P}(\bar{\theta}_n' = \theta' | \theta_n = \theta) \\
&= Q_n(\theta, \theta') \mathbb{P}(U_n \leq \exp(-L'/T_n)) \\
&\geq q \exp(-L'/T_n).
\end{aligned}$$

The first inequality above follows from equation (B.1) and the fact a candidate solution only depends on a current iterate. The third equality holds because U_n is independent of everything else. The final inequality follows by the choice of n and the fact that U_n is a $U[0, 1]$ random variable. ■

In the next lemma, we present a bound on the one-step transition probability from the current point θ to itself for $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$, provided that $\theta \in \Theta \setminus \Theta_L$.

Lemma B.2. *Suppose that Assumptions 4.1, 4.2, 4.11, and 4.12 are satisfied. Then, for all $\theta \in \Theta \setminus \Theta_L$, there exists $n_1 \in \mathbb{N}$, $n_1 \geq n_0$, such that for all $n \geq (n_1 - 1)r$, we have that*

$$\mathbb{P}(\bar{\theta}_{n+1} = \theta | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \geq q \exp(-L'/T_n),$$

provided that $k \leq m \leq n$, $k_i' \geq k(\omega_s)$ *for* $i = 1, \ldots, b$, *and* $\mathbb{P}(A(m, n, \theta, \{k_i'\}_{i=1}^b)) > 0$.

Proof: Fix $\theta \in \Theta \setminus \Theta_L$. Now observe that if $n \geq k$, then for all $\omega_d \in \Omega_d$, we have that

$$\begin{aligned}
[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\gamma_\theta)]^+ &= [\tilde{f}_{\bar{C}_{n+1}(\theta)}(\theta, \omega_s) - \tilde{f}_{\bar{C}_{n+1}(\gamma_\theta)}(\gamma_\theta, \omega_s)]^+ \\
&\geq [f(\theta) - \epsilon - f(\gamma_\theta) - \epsilon]^+ \\
&\geq [3\epsilon - 2\epsilon]^+ \\
&= \epsilon. \qquad \text{(B.2)}
\end{aligned}$$

The first inequality follows from the fact that $\bar{C}_{n+1}(\theta), \bar{C}_{n+1}(\gamma_\theta) \geq k(\omega_s)$, while the second one follows from the definitions of α and ϵ.

For $n \geq (n_0 - 1)r$, we have that

$$\begin{aligned}
&\mathbb{P}(\bar{\theta}_{n+1} = \theta | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}\left(\bigcup_{\theta' \in N_n(\theta)} \{\bar{\theta}_n' = \theta', U_n > \exp(-[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\theta')]^+ / T_n)\} \,\middle|\, A(m, n, \theta, \{k_i'\}_{i=1}^b) \right) \\
&\geq \mathbb{P}(\bar{\theta}_n' = \gamma_\theta, U_n > \exp(-[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\gamma_\theta)]^+ / T_n) | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}(U_n > \exp(-[\bar{f}_{n+1}(\theta) - \bar{f}_{n+1}(\gamma_\theta)]^+ / T_n) | A(m, n, \theta, \{k_i'\}_{i=1}^b), \bar{\theta}_n' = \gamma_\theta) \\
&\quad \times \mathbb{P}(\bar{\theta}_n' = \gamma_\theta | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \\
&\geq \mathbb{P}(U_n > \exp(-\epsilon / T_n) | A(m, n, \theta, \{k_i'\}_{i=1}^b), \bar{\theta}_n' = \gamma_\theta) \mathbb{P}(\bar{\theta}_n' = \gamma_\theta | \theta_n = \theta) \\
&= Q_n(\theta, \gamma_\theta) \mathbb{P}(U_n > \exp(-\epsilon / T_n)) \\
&\geq q(1 - \exp(-\epsilon / T_n)). \qquad \text{(B.3)}
\end{aligned}$$

The second inequality above follows from equation (B.2) and the fact a candidate solution only depends on a current iterate. The third equality holds because U_n is independent of everything else. The final inequality follows from Assumption 4.11, the choice of n, and the fact that U_n is a $U[0, 1]$ random variable.

Assumption 4.12 and equation (B.3) imply that there exists $n(\theta) \geq n_0$ such that $n \geq (n(\theta) - 1)r$ implies that $\exp(-L'/T_n) \leq 1/2$ and $\mathbb{P}(\bar{\theta}_{n+1} = \theta | A(m, n, \theta)) \geq q/2$. Therefore, if $n \geq (n(\theta) - 1)r$, then

$$\mathbb{P}(\bar{\theta}_{n+1} = \theta | A(m, n, \theta, \{k_i'\}_{i=1}^b)) \geq q \exp(-L'/T_n).$$

Let $n_1 = \max_{\theta \in \Theta \setminus \Theta_L} n(\theta)$. Assumption 4.2 insures that $n_1 \in \mathbb{N}$, and the proof is complete. ■

Next we derive a lower bound on the value of the r-step transition probability between any two feasible points for $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$ for a sufficiently large starting iteration number.

Lemma B.3. *Suppose that Assumptions 4.1, 4.2, 4.11, 4.12, and 4.14 hold. Then for all $\theta, \theta' \in \Theta$ and $n \geq n_1 r$, we have that*

$$\mathbb{P}(\bar{\theta}_n = \theta' | A(m, n-r, \theta, \{k_i'\}_{i=1}^b)) \geq q^r \exp(-rL'/T_{n-1}),$$

provided that $k \leq m \leq n-r$, $k_i' \geq k(\omega_s)$ for $i = 1, \ldots, b$, and $\mathbb{P}(A(m, n-r, \theta, \{k_i'\}_{i=1}^b)) > 0$.

Proof: Let $x_{n-r} = \theta, \ldots, x_n = \theta' \in \Theta$ be such that either $x_{n-r+i} \in N(x_{n-r+i-1})$ or $x_{n-r+i} = x_{n-r+i-1} \in \Theta \setminus \Theta_L$ for $i = 1, \ldots, r$ (note that this is possible by the definition of r). We first verify that $\mathbb{P}(A(m, n-r+j, x_{n-r+j}, \{k_i'\}_{i=1}^b)) > 0$ for all $j = 0, \ldots, r$. The statement is obviously correct for $j = 0$. Suppose that it is correct for $0 \leq j < r$. Then, by conditioning on $A(m, n-r+j, x_{n-r+j}, \{k_i'\}_{i=1}^b)$, we obtain that

$$\begin{aligned}
&\mathbb{P}(A(m, n-r+j+1, x_{n-r+j+1}, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}(A(m, n-r+j+1, x_{n-r+j+1}, \{k_i'\}_{i=1}^b) | A(m, n-r+j, x_{n-r+j}, \{k_i'\}_{i=1}^b)) \\
&\quad \times \mathbb{P}(A(m, n-r+j, x_{n-r+j}, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}(\bar{\theta}_{n-r+j+1} = x_{n-r+j+1} | A(m, n-r+j, x_{n-r+j}, \{k_i'\}_{i=1}^b)) \\
&\quad \times \mathbb{P}(A(m, n-r+j, x_{n-r+j}, \{k_i'\}_{i=1}^b)) \\
&> 0. \qquad \text{(B.4)}
\end{aligned}$$

The inequality follows from the induction hypothesis and Lemmas B.1 and B.2. This proves the claim.

By conditioning on $\bar{\theta}_{n-r+1}$ we obtain that

$$\begin{aligned}
\mathbb{P}(\bar{\theta}_n = \theta' | A(m, n-r, \theta, \{k_i'\}_{i=1}^b)) &= \sum_{y \in \Theta} \mathbb{P}(\bar{\theta}_n = \theta' | A(m, n-r, \theta, \{k_i'\}_{i=1}^b), \bar{\theta}_{n-r+1} = y) \\
&\quad \times \mathbb{P}(\bar{\theta}_{n-r+1} = y | A(m, n-r, \theta, \{k_i'\}_{i=1}^b)) \\
&\geq \mathbb{P}(\bar{\theta}_n = \theta' | A(m, n-r, \theta, \{k_i'\}_{i=1}^b), \bar{\theta}_{n-r+1} = x_{n-r+1}) \\
&\quad \times \mathbb{P}(\bar{\theta}_{n-r+1} = x_{n-r+1} | A(m, n-r, \theta, \{k_i'\}_{i=1}^b)) \\
&= \mathbb{P}(\bar{\theta}_n = \theta' | A(m, n-r+1, x_{n-r+1}, \{k_i'\}_{i=1}^b)) \\
&\quad \times \mathbb{P}(\bar{\theta}_{n-r+1} = x_{n-r+1} | A(m, n-r, \theta, \{k_i'\}_{i=1}^b)).
\end{aligned}$$

Thus, proceeding iteratively in a similar manner, we obtain that

$$\begin{aligned}
&\mathbb{P}(\bar{\theta}_n = \theta' | A(m, n-r, \theta, \{k_i'\}_{i=1}^b)) \\
&\quad \geq \prod_{j=1}^{r} \mathbb{P}(\bar{\theta}_{n-r+j} = x_{n-r+j} | A(m, n-r+j-1, x_{n-r+j-1}, \{k_i'\}_{i=1}^b)) \\
&\quad \geq \prod_{j=1}^{r} (q \exp(-L'/T_{n-r+j-1})) \\
&\quad \geq q^r \exp(-rL'/T_{n-1}).
\end{aligned}$$

The second inequality follows from Lemmas B.1 and B.2 and equation (B.4), while the last inequality follows from Assumption 4.12. ■

Define $\Omega_k^\infty = \prod_{n=k}^\infty \Omega_n$ and l to be the smallest integer such that $lr \geq k$. For $\theta \in \Theta$ and $n \in \mathbb{N}$ such that $n \geq l+1$, let $A_\theta^n = \{\omega_d \in \Omega_k^\infty : \bar{\theta}_{nr}(\omega_d) = \theta\}$. Also, for each $n \in \mathbb{N}$ such that $n \geq l$, let $\mathcal{F}_n$ be the σ-algebra generated by $\{\bar{\theta}_j\}_{j=k}^{nr}$. Observe that $A_\theta^n \in \mathcal{F}_n$. The next lemma provides a sufficient condition on the cooling schedule which ensures that each $\theta \in \Theta$ is visited infinitely often (i.o.) with probability one by $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$.

Lemma B.4. *Suppose that Assumptions 4.1, 4.2, 4.11, 4.12, and 4.14 hold. Then, for each $\theta \in \Theta$, $\mathbb{P}(A_\theta^n$ i.o.$) = 1$, provided that the cooling schedule satisfies equation (4.6) and $k_i \geq k(\omega_s)$ for $i = 1, \ldots, b$.*

Proof: Fix $\theta \in \Theta$. Let $n \geq n_1$ and fix any $\{x_j\}_{j=k}^{(n-1)r} \subset \Theta$ such that $\mathbb{P}(A(k, (n-1)r, x_{(n-1)r}, \{k_i\}_{i=1}^b)) > 0$ (such a $\{x_j\}_{j=k}^{(n-1)r}$ exists because there are only finitely many

such paths under Assumption 4.2, and hence $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$ must follow at least one of them with positive probability). Then we have that

$$\begin{aligned}\mathbb{P}(\bar{\theta}_{nr} = \theta | \bar{\theta}_k = x_k, \ldots, \bar{\theta}_{(n-1)r} = x_{(n-1)r}) &= \mathbb{P}(\bar{\theta}_{nr} = \theta | A(k, (n-1)r, x_{(n-1)r}, \{k_i\}_{i=1}^b)) \\ &\geq q^r \exp(-rL'/T_{nr-1}). \end{aligned} \tag{B.5}$$

The equality follows from the fact that the additional information on which we condition on the right-hand side is a probability one set. The inequality follows from Lemma B.3.

Now observe that, for $n \geq n_1$, we have for almost every $\omega_d \in \Omega_d$ that

$$\begin{aligned}&\mathbb{P}(A_\theta^n | \mathcal{F}_{n-1}) \\ &= \sum_{x_k, \ldots, x_{(n-1)r} \in \Theta} \mathbb{P}(\bar{\theta}_{nr} = \theta | \bar{\theta}_k = x_k, \ldots, \bar{\theta}_{(n-1)r} = x_{(n-1)r}) I(\bar{\theta}_k = x_k, \ldots, \bar{\theta}_{(n-1)r} = x_{(n-1)r}) \\ &\geq q^r \exp(-rL'/T_{nr-1}), \end{aligned} \tag{B.6}$$

where $I(\cdot)$ is an indicator function. The inequality follows from the fact that equation (B.5) holds whenever $\{\bar{\theta} = x_k, \ldots, \bar{\theta}_{(n-1)r} = x_{(n-1)r}\}$ occurs with positive probability. Then we have for almost every $\omega_d \in \Omega_d$ that

$$\sum_{n=l+1}^{\infty} \mathbb{P}(A_\theta^n | \mathcal{F}_{n-1}) \geq \sum_{n=n_1}^{\infty} \mathbb{P}(A_\theta^n | \mathcal{F}_{n-1}) \geq \sum_{n=n_1}^{\infty} q^r \exp(-rL'/T_{nr-1}) = +\infty.$$

The second inequality follows by (B.6). The final equality follows from equation (4.6) and Lemma 4.3 (note that Lemma 4.3 is still valid with L substituted by L'). Consequently, we conclude by the conditional Borel-Cantelli lemma (see, e.g., Corollary 2.3 on page 32 in Hall and Heyde [43]) that $\mathbb{P}(A_\theta^n \text{ i.o.}) = 1$. ∎

We now proceed to construct the subset $\tilde{\Omega}_d(\omega_s)$ of Ω_d. For each $x \in \Theta$, $k \in \mathbb{N}$, and $\{k_i\}_{i=1}^b \subset \mathbb{N}$ such that $k_i \geq k(\omega_s)$ for $i = 1, \ldots, b$, define

$$A_k(\omega_s, x, \{k_i\}_{i=1}^b) = \left\{ \omega_d \in \Omega_k^\infty : Alg(\omega_s, x, k, \{k_i\}_{i=1}^b) \text{ samples each } \theta' \in \Theta \text{ i.o.} \right\}.$$

In other words,

$$A_k(\omega_s, x, \{k_i\}_{i=1}^b) = \bigcap_{\theta' \in \Theta} \{\omega_d \in \Omega_k^\infty : \theta' \in \bar{\Theta}_n \text{ i.o.}\},$$

where $\bar{\Theta}_n = \{\bar{\theta}_n, \bar{\theta}'_n\}$ for all $n \geq k$. Lemma B.4 and Assumption 4.2 give that $\mathbb{P}(A_k(\omega_s, x, \{k_i\}_{i=1}^b))=1$. For each $k \in \mathbb{N}$, let

$$A_k(\omega_s) = \bigcap_{x \in \Theta} \bigcap_{k_i \geq k(\omega_s), i=1,\ldots,b} A_k(\omega_s, x, \{k_i\}_{i=1}^b).$$

Assumption 4.2 ensures that the intersection above is taken over countably many sets. Thus, we have that $\mathbb{P}(A_k(\omega_s)) = 1$ for all $k \in \mathbb{N}$. For each $k \in \mathbb{N}$, let $\tilde{A}_k(\omega_s) = (\prod_{n=0}^{k-1} \Omega_n) \times A_k(\omega_s)$. Obviously $\mathbb{P}(\tilde{A}_k(\omega_s)) = 1$. Finally, let $\tilde{\Omega}_d(\omega_s) = \bigcap_{k=0}^{\infty} \tilde{A}_k(\omega_s)$, the set of all sample paths having the feature that for all $x \in \Theta$, $k \in \mathbb{N}$, and $\{k_i\}_{i=1}^b \in \mathbb{N}$ such that $k_i \geq k(\omega_s)$ for $i = 1, \ldots, b$, $Alg(\omega_s, x, k, \{k_i\}_{i=1}^b)$ samples each feasible point infinitely often. Clearly, $\mathbb{P}(\tilde{\Omega}_d(\omega_s)) = 1$.

Fix $\omega \in (\tilde{\Omega}_d(\omega_s) \times \{\omega_s\}) \cap \bar{\Omega}$. We next show that Algorithm 4.4 samples each feasible solution infinitely often under this ω. We proceed by contradiction. Let

$$\bar{\Theta}(\omega) = \{\theta \in \Theta : \text{ Algorithm 4.4 samples } \theta \text{ i.o. under } \omega\}.$$

Suppose that $\bar{\Theta}(\omega) \neq \Theta$. Then by Assumption 4.2 and the choice of ω, there exists an iteration number $n_2(\omega)$ such that $n \geq n_2$ implies that $\Theta_n \subset \bar{\Theta}(\omega)$ and $C_n(\theta, \omega) \geq k(\omega_s)$ for all $\theta \in \bar{\Theta}(\omega)$. Observe that due to Assumption 4.16, Algorithm 4.4 under this ω couples with $Alg(\omega_s, \theta_{n_2}, n_2, \{k_i\}_{i=1}^b)$, where $k_i = C_{n_2}(i, \omega) \geq k(\omega_s)$ for all $i \in \bar{\Theta}(\omega)$ and $k_i = k(\omega_s)$ for $i \notin \bar{\Theta}(\omega)$, even though $C_n(i, \omega)$ and $\bar{C}_n(i, \omega)$ may not agree for $i \in \bar{\Theta}(\omega)$ (because the decisions made in every iteration depend only on the information collected so far at the current and candidate solutions, and this information agrees for Algorithm 4.4 and $Alg(\omega_s, \theta_{n_2}, n_2, \{k_i\}_{i=1}^b)$ after iteration n_2). But by the choice of ω, we know that $Alg(\omega_s, \theta_{n_2}, n_2, \{k_i\}_{i=1}^b)$ samples each feasible point infinitely often. This provides a contradiction, and hence we have shown that Algorithm 4.4 samples all $\theta \in \Theta$ infinitely often under this ω.

Let $\beta = \frac{1}{2}\left[\max_{\theta \in \Theta} f(\theta) - \max_{\theta \in \Theta \setminus \Theta^*} f(\theta)\right]$, which is strictly positive under Assumption 4.14. Then by Assumption 4.2 and the choice of ω, there exists $N(\omega) \in \mathbb{N}$ such that for all $\theta \in \Theta$ and $n \geq N(\omega)$, we have that $|\hat{f}_n(\theta) - f(\theta)| < \beta$. This shows that $\theta_n^* \in \Theta^*$ for all $n \geq N(\omega)$.

Let $\tilde{\Omega} = \cup_{\omega_s \in \tilde{\Omega}_s} \tilde{\Omega}_d(\omega_s) \times \{\omega_s\}$. Observe that $\tilde{\Omega} \cap \bar{\Omega}$ is a subset of Ω under which Algorithm 4.4 converges to Θ^*. Hence it suffices to verify that $\mathbb{P}(\tilde{\Omega}) = 1$ because $\mathbb{P}(\bar{\Omega}) = 1$. We have that

$$\begin{aligned}
\mathbb{P}(\tilde{\Omega}) &= \int_{\Omega} I\{(\omega_d, \omega_s) \in \tilde{\Omega}\} \, \mathrm{d}\mathbb{P}(\omega_d, \omega_s) \\
&= \int_{\Omega_s} \left(\int_{\Omega_d} I\{(\omega_d, \omega_s) \in \tilde{\Omega}\} \, \mathrm{d}\mathbb{P}_d(\omega_d) \right) \mathrm{d}\mathbb{P}_s(\omega_s) \\
&\geq \int_{\tilde{\Omega}_s} \left(\int_{\Omega_d} I\{(\omega_d, \omega_s) \in \tilde{\Omega}\} \, \mathrm{d}\mathbb{P}_d(\omega_d) \right) \mathrm{d}\mathbb{P}_s(\omega_s) \\
&= \int_{\tilde{\Omega}_s} \mathbb{P}_d(\tilde{\Omega}_d(\omega_s)) \, \mathrm{d}\mathbb{P}_s(\omega_s) \\
&= \int_{\tilde{\Omega}_s} 1 \, \mathrm{d}\mathbb{P}_s(\omega_s) \\
&= \mathbb{P}_s(\tilde{\Omega}_s) \\
&= 1.
\end{aligned}$$

The second equality follows by Assumption 4.1 and Fubini's theorem. This concludes the proof of Theorem 4.5. ■

Remark B.1. The conclusion of Theorem 4.5 is still valid when the first part of Assumption 4.16 is substituted by the statement that $K_n(\theta_n)$ and $K_n(\theta'_n)$ depend only on all the objective function observations collected by the algorithm in the first $n-1$ iterations at each $\theta \in N(\theta_n)$, $\{C_n(\theta)\}_{\theta \in N(\theta_n)}$, and the iteration number n. The proof of this result is more complicated and involves more notation than the proof of Theorem 4.5. We did not present the proof of this extension of Theorem 4.5 because we think that our proof ideas are a more substantial contribution than this extension of Theorem 4.5, and that these ideas can be better comprehended via a proof that is less notational. However, the main idea behind the proof of this extension is to specify Ω_d in a way that allows us to identify a probability one subset $\tilde{\Omega}_d(\omega_s)$ of Ω_d under which the SA algorithm with averaging not only visits each feasible solution infinitely often, but also all neighbors $\theta' \in N(\theta)$ of each feasible solution $\theta \in \Theta$ will be chosen as candidate solutions infinitely often when θ is a current solution, provided that the optimization method is initialized with a "sufficient" number of observations collected at each point and these observations are collected under ω_s. The remaining parts of the proof use similar ideas as those of the proof of Theorem 4.5.

APPENDIX C

PROOF OF LEMMA 5.2

In order to prove Lemma 5.2, we need the following result that generalizes Lemma 2.3 in Baumert and Smith [20].

Lemma C.1. *Let $p, q \in (0,1)$ be such that $p+q<1$. Let $n_k = O(k^p)$ and $L_k = O(k^q)$ be sequences of positive integers and $V_k = \Phi(k^{-p})$ be a sequence of positive real numbers. Suppose that for all $k \geq 1$ and $j = 1, \ldots, n_k$, A_k^j is a subset of Θ such that $G(A_k^j) \geq V_k$. For each $k \in \mathbb{N}$, let $E_k = \cap_{j=1}^{n_k}\{N_k(A_k^j) \geq L_k\}$. Then $\sum_{k=1}^{\infty}\mathbb{P}(\bar{E}_k) < \infty$.*

Proof: Note that $\bar{E}_k = \cup_{j=1}^{n_k}\bar{E}_k^j$, where E_k^j is the event $\{N_k(A_k^j) \geq L_k\}$. Observe that $N_k(A_k^j)$ is a $Bin(k, G(A_k^j))$ random variable. Since $V_k = \Phi(k^{-p})$, there is a $C \in \mathbb{R}^+$ such that for all $k \geq 1$ and $j = 1, \ldots, n_k$, $G(A_k^j) \geq 1/(Ck^p)$. Moreover, the probability that there are n or fewer "successes" in k Bernoulli trials is maximized when the probability of a "success" is minimized. Hence, if k satisfies $L_k \leq k$, we have that

$$\mathbb{P}(\bar{E}_k) \leq \sum_{j=1}^{n_k}\mathbb{P}(Bin(k, 1/(Ck^p)) \leq L_k - 1) = n_k \sum_{n=0}^{L_k-1}\binom{k}{n}\left(\frac{1}{Ck^p}\right)^n\left(1-\frac{1}{Ck^p}\right)^{k-n}. \quad \text{(C.1)}$$

Let k be sufficiently large so that $L_k \leq k/2$ and $Ck^p \geq 3$. Suppose also that $L_k \leq Lk^q$ for all $k \in \mathbb{N}$. Then equation (C.1) implies that

$$\begin{aligned}
\mathbb{P}(\bar{E}_k) &\leq n_k\left(\frac{1}{Ck^p}\right)\sum_{n=0}^{L_k-1}\binom{k}{n}\left(\frac{1}{Ck^p}\right)^{n-1}\left(1-\frac{1}{Ck^p}\right)^{k-n} \\
&\leq const \times \binom{k}{L_k}\sum_{n=0}^{L_k-1}\frac{(Ck^p-1)^{k-n}}{(Ck^p)^{k-1}} \qquad \text{(C.2)} \\
&\leq const \times k^{L_k}\frac{(Ck^p-1)^k}{(Ck^p)^{k-1}}\sum_{n=0}^{L_k-1}\left(\frac{1}{Ck^p-1}\right)^n \qquad \text{(C.3)} \\
&\leq const \times k^{L_k}\frac{(Ck^p-1)^k}{(Ck^p)^{k-1}} \times \frac{Ck^p-1}{Ck^p-2} \qquad \text{(C.4)} \\
&\leq const \times k^{Lk^q+2}\left(1-\frac{1}{Ck^p}\right)^k.
\end{aligned}$$

Equation (C.2) follows by the fact that $n_k = O(k^p)$ and $\binom{k}{n} \leq \binom{k}{L_k}$ when $L_k \leq k/2$ and $n \leq L_k$. Equation (C.3) follows from the fact that $\binom{k}{n} \leq k^n$. Equation (C.4) follows because

$$\sum_{n=0}^{L_k-1} \left(\frac{1}{Ck^p-1}\right)^n \leq \sum_{n=0}^{\infty} \left(\frac{1}{Ck^p-1}\right)^n = \frac{Ck^p-1}{Ck^p-2}.$$

The last inequality follows by algebra.

Observe that there exists $0 \leq \alpha < 1$ such that for large k, we have that $(1-1/(Ck^p))^{k^p} \leq \alpha$. Thus, for all large k, we have that $\mathbb{P}(\bar{E}_k) \leq const \times k^{Lk^q+2}\alpha^{k^{1-p}}$. Let $\epsilon > 0$ be such that $q + \epsilon < 1 - p$. Since $\ln(k^{Lk^q+2})/k^{q+\epsilon} \to 0$ as $k \to \infty$, we have that $k^{Lk^q+2} \leq \exp(k^{q+\epsilon})$ for large k. By the choice of ϵ and the fact that $\alpha < 1$, we have that for large k,

$$\mathbb{P}(\bar{E}_k) \leq const \times \exp(k^{q+\epsilon} + \ln(\alpha)k^{1-p}) \leq const \times \exp(-k^{q+\epsilon}).$$

Observe that $\exp(-x^t)$ is a decreasing function in x when $0 < t < 1$ with $\int_0^\infty \exp(-x^t)\mathrm{d}x < \infty$ (this follows by the change of variable $y = x^t$ and the fact that an exponential random variable has all moments finite). The result now follows by the integral test. ■

Proof of Lemma 5.2: Because $r_k \to 0$ as $k \to \infty$, there exists $k' \in \mathbb{N}$ such that $r_k \leq \bar{\epsilon}$ for all $k \geq k'$, where $\bar{\epsilon}$ is the strictly positive constant of Assumption 1.1 in Baumert and Smith [20]. Note that it suffices to verify that $\sum_{k=k'}^{\infty} \mathbb{P}(\bar{D}_k) < \infty$. Suppose that for each $k \geq k'$, we partition each dimension of $\mathbb{R}^s$ into segments of length $r_k/(3\sqrt{s})$, and by doing so we obtain closed subsets of $\mathbb{R}^s$ that cover $\mathbb{R}^s$. We refer to each such subset of $\mathbb{R}^s$ as a grid box. For each $\theta \in \Theta$, let T_θ be some grid box containing $\theta \in \Theta$. Define H_θ as the union of T_θ and all the grid boxes adjacent T_θ (two grid boxes are adjacent if their intersection is not empty). Assumption 5.7 ensures that H_θ covers all points that are at most $r_k/(3\sqrt{s})$ from T_θ, and hence we have that $B(\theta, r_k/(3\sqrt{s})) \subset H_\theta$. Thus,

$$G(H_\theta \cap \Theta) \geq G(B(\theta, r_k/(3\sqrt{s}))) \geq C_1 \times \mathcal{L}(\tilde{B}(\theta, r_k/(3\sqrt{s}))) = C_2 \times (r_k)^s = \Phi(k^{-p}), \quad \text{(C.5)}$$

where $\mathcal{L}$ is the Lebesgue measure on $\mathbb{R}^s$, $\tilde{B}(\theta, r) = \{x \in \mathbb{R}^s : d(x, \theta) \leq r\}$, and C_1, C_2 are positive constants. The second inequality follows from Assumptions 5.6 and 5.7 and Lemma 2.4 in Baumert and Smith [20]. The final equality follows from the fact that $r_k = \Phi(k^{-p/s})$.

Because Θ is bounded by Assumption 5.6, each dimension needs to be partitioned into a $O(k^{p/s})$ segments (recall that the increments are $\Phi(k^{-p/s})$). Thus, the total number

of grid boxes T_θ necessary to cover Θ is $O(k^p)$. Because $T_\theta \subset H_\theta$ for each $\theta \in \Theta$, we conclude that Θ can be covered with $n_k = O(k^p)$ H_θ sets. Now consider a collection of sets $H_{\theta_1}, \ldots, H_{\theta_{n_k}}$, that covers Θ. Let E_k be the event of Lemma C.1 with $A_k^1, \ldots, A_k^{n_k}$ given by the collection $H_{\theta_1} \cap \Theta, \ldots, H_{\theta_{n_k}} \cap \Theta$, covering Θ. By Lemma C.1 and equation (C.5), we have that $\sum_{k=k'}^{\infty} \mathbb{P}(\bar{E}_k) < \infty$. For each $k \geq k'$ and $\theta \in \Theta$, we can find $1 \leq i \leq n_k$ such that $H_{\theta_i} \cap \{\theta\} = \{\theta\}$ (because $H_{\theta_1}, \ldots, H_{\theta_{n_k}}$ cover Θ). Because the maximum distance between any two points in H_{θ_i} is r_k, we get that $H_{\theta_i} \subset \tilde{B}(\theta, r_k)$, and hence that $H_{\theta_i} \cap \Theta \subset B(\theta, r_k)$. This shows that $E_k \subset D_k$ for all $k \geq k'$ and the proof is complete. ■

REFERENCES

[1] Ahmed, M. A. and Alkhamis, T. M., "Simulation-based optimization using simulated annealing with ranking and selection," *Computers and Operations Research*, vol. 29, no. 4, pp. 383–402, 2002.

[2] Alexander, D. L. J., Bulger, D. W., Calvin, J. M., Romejin, H. E., and Sherriff, R. L., "Approximate implementations of pure random search in the presence of noise," *Journal of Global Optimization*, vol. 31, no. 4, pp. 601–612, 2005.

[3] Alrefaei, M. H. and Andradóttir, S., "A simulated annealing algorithm with constant temperature for discrete stochastic optimization," *Management Science*, vol. 45, no. 5, pp. 748–764, 1999.

[4] Alrefaei, M. H. and Andradóttir, S., "A modification of the stochastic ruler method for discrete stochastic optimization," *European Journal of Operational Research*, vol. 133, no. 1, pp. 160–182, 2001.

[5] Alrefaei, M. H. and Andradóttir, S., "Discrete stochastic optimization using variants of the stochastic ruler method," *Naval Research Logistics*, vol. 52, no. 4, pp. 344–360, 2005.

[6] Andradóttir, S., "A method for discrete stochastic optimization," *Management Science*, vol. 41, no. 12, pp. 1946–1961, 1995.

[7] Andradóttir, S., "A stochastic approximation algorithm with varying bounds," *Operations Research*, vol. 43, no. 6, pp. 1037–1048, 1995.

[8] Andradóttir, S., "A global search method for discrete stochastic optimization," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 513–530, 1996.

[9] Andradóttir, S., "A scaled stochastic approximation algorithm," *Management Science*, vol. 42, no. 4, pp. 475–498, 1996.

[10] Andradóttir, S., "Simulation optimization," in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice* (Banks, J., ed.), pp. 307–333, New York: Wiley, 1998.

[11] Andradóttir, S., "Accelerating the convergence of random search methods for discrete stochastic optimization," *ACM Transactions on Modeling and Computer Simulation*, vol. 9, no. 4, pp. 349–380, 1999.

[12] Andradóttir, S., "Rate of convergence of random search methods for discrete optimization using steady-state simulation." Preprint, 2000.

[13] Andradóttir, S., "An overview of simulation optimization via random search," in *Handbooks in Operations Research and Management Science: Simulation* (Henderson, S. G. and Nelson, B. L., eds.), pp. 617–632, Amsterdam: Elsevier Science, 2006.

[14] Andradóttir, S., "Simulation optimization with countably infinite feasible regions: Efficiency and convergence," *ACM Transactions on Modeling and Computer Simulation*, vol. 16, no. 4, pp. 357–374, 2006.

[15] Andradóttir, S. and Kim, S.-H., "Fully sequential procedures for comparing constrained systems via simulation." Submitted paper, 2007.

[16] April, J., Better, M., Glover, F., and Kelly, J., "New advances and applications for marrying simulation and optimization," in *Proceedings of the 2004 Winter Simulation Conference* (Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., eds.), pp. 80–86, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2004.

[17] Azadivar, F., Shu, J., and Ahmad, M., "Simulation optimization in strategic location of semi-finished products in a pull-type production system," in *Proceedings of the 1996 Winter Simulation Conference* (Charnes, J. M., Morrice, D. J., Brunner, D. T., and Swain, J. J., eds.), pp. 1123–1128, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 1996.

[18] Banks, J. A., Carson, J. S., Nelson, B. L., and Nicol, D. M., *Discrete-Event System Simulation*. Upper Saddle River, NJ: Prentice Hall, 4th ed., 2004.

[19] Batur, D. and Kim, S.-H., "Fully sequential selection procedures with parabolic boundary," *IIE Transactions*, vol. 38, no. 9, pp. 749–764, 2006.

[20] Baumert, S. and Smith, R. L., "Pure random search for noisy objective functions," tech. rep., University of Michigan, May 2002.

[21] Benveniste, A., Métivier, M., and Priouret, P., *Adaptive Algorithms and Stochastic Approximations*. Berlin: Springer-Verlag, 1990.

[22] Bhatnagar, S. and Borkar, V. S., "A two time scale stochastic approximation scheme for simulation based parametric optimization," *Probability in the Engineering and Informational Sciences*, vol. 12, pp. 519–531, 1998.

[23] Blomvall, J. and Shapiro, A., "Solving multistage asset investment problems by the sample average approximation method," *Mathematical Programming*, vol. 108, no. 2-3, pp. 571–595, 2006.

[24] Boesel, J., Nelson, B. L., and Ishii, N., "A framework for simulation-optimization software," *IIE Transactions*, vol. 35, no. 1, pp. 221–229, 2003.

[25] Boesel, J., Nelson, B. L., and Kim, S.-H., "Using ranking and selection to clean up after a simulation search," *Operations Research*, vol. 51, no. 5, pp. 814–825, 2003.

[26] Buzacott, J. A. and Shantikumar, J. G., *Stochastic Models of Manufacturing Systems*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[27] Carson, Y. and Maria, A., "Simulation optimization: Methods and applications," in *Proceedings of the 1997 Winter Simulation Conference* (Andradóttir, S., Healy, K. J., Withers, D. H., and Nelson, B. L., eds.), pp. 118–126, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 1997.

[28] Chung, K. L., *A Course in Probability Theory.* San Diego, CA: Academic Press, Third ed., 2001.

[29] Dai, L. Y., "Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems," *Journal of Optimization Theory and Applications*, vol. 91, no. 2, pp. 363–388, 1996.

[30] Dai, L. Y. and Chen, C.-H., "Rates of convergence of ordinal comparison for dependent discrete event dynamic systems," *Journal of Optimization Theory and Applications*, vol. 94, no. 1, pp. 29–54, 1997.

[31] Fox, B. L. and Heine, G. W., "Probabilistic search with overrides," *The Annals of Applied Probability*, vol. 5, no. 4, pp. 1087–1094, 1995.

[32] Fu, M. C., "Optimization for simulation: Theory vs. practice," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.

[33] Fu, M. C., "Gradient estimation," in *Handbooks in Operations Research and Management Science: Simulation* (Henderson, S. G. and Nelson, B. L., eds.), pp. 575–616, Amsterdam: Elsevier Science, 2006.

[34] Fu, M. C. and Healy, K. J., "Simulation optimization of (s,S) inventory systems," in *Proceedings of the 1992 Winter Simulation Conference* (Swain, J. J., Goldsman, D., Chain, R. C., and Wilson, J. R., eds.), pp. 506–514, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 1992.

[35] Fu, M. C. and Hu, J. Q., *Conditional Monte Carlo: Gradient Estimation and Optimization Applications.* Norwell, MA: Kluwer, 1997.

[36] Gelfand, S. B. and Mitter, S. K., "Simulated annealing with noisy or imprecise energy measurements," *Journal of Optimization Theory and Applications*, vol. 62, no. 1, pp. 49–62, 1989.

[37] Ghate, A. and Smith, R. L., "Adaptive search with stochastic acceptance probabilities for global optimization." Submitted paper, 2007.

[38] Glasserman, P., *Gradient Estimation via Perturbation Analysis.* Boston, MA: Kluwer Academic Publishers, 1991.

[39] Goldsman, D. and Nelson, B. L., "Comparing systems via simulation," in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice* (Banks, J., ed.), pp. 273–306, New York: Wiley, 1998.

[40] Gong, W.-B., Ho, Y.-C., and Zhai, W., "Stochastic comparison algorithm for discrete optimization with estimation," *SIAM Journal on Optimization*, vol. 10, no. 2, pp. 384–404, 1999.

[41] Gutjahr, W. J. and Pflug, G. C., "Simulated annealing for noisy cost functions," *Journal of Global Optimization*, vol. 8, no. 1, pp. 1–13, 1996.

[42] Hajek, B., "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, 1988.

[43] Hall, P. and Heyde, C. C., *Martingale Limit Theory and its Applications*. New York, NY: Academic Press, 1980.

[44] Healy, K. J. and Schruben, L. W., "Retrospective simulation response optimization," in *Proceedings of the 1991 Winter Simulation Conference* (Nelson, B. L., Kelton, W. D., and Clark, G. M., eds.), pp. 901–906, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 1991.

[45] Hill, S. D. and Fu, M. C., "Simulation optimization via simultaneous perturbation stochastic approximation," in *Proceedings of the 1994 Winter Simulation Conference* (Tew, J. D., Manivannan, S., Sadowski, D. A., and Seila, A. F., eds.), pp. 1461–1464, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 1994.

[46] Ho, Y.-C., "An explanation of ordinal optimization: Soft computing for hard problems," *Discrete Event Dynamic Systems*, vol. 2, pp. 61–88, 1992.

[47] Ho, Y.-C. and Cao, X.-R., *Perturbation Analysis of Discrete Event Dynamical Systems*. Norwell, MA: Kluwer, 1991.

[48] Ho, Y.-C., Sreenivas, S., and Vakili, P., "Ordinal optimization of DEDS," *Discrete Event Dynamic Systems*, vol. 2, pp. 61–88, 1992.

[49] Homem-de-Mello, T., "Variable-sample methods for stochastic optimization," *ACM Transactions on Modeling and Computer Simulation*, vol. 13, no. 2, pp. 108–133, 2003.

[50] Hong, L. J. and Nelson, B. L., "The tradeoff between sampling and switching: New sequential procedures for indifference-zone selection," *IIE Transactions*, vol. 37, no. 7, pp. 623–634, 2005.

[51] Hong, L. J. and Nelson, B. L., "Discrete optimization via simulation using COMPASS," *Operations Research*, vol. 54, no. 1, pp. 115–129, 2006.

[52] Hong, L. J., "Discrete optimization via simulation using coordinate search," in *Proceedings of the 2005 Winter Simulation Conference* (Kuhl, M. E., Steiger, N. M., Armstrong, F. B., and Joines, J. A., eds.), pp. 803–810, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2005.

[53] Hu, J., Fu, M. C., and Marcus, S. I., "A model reference adaptive search algorithm for stochastic global optimization." Submitted paper, August 2006.

[54] Kim, S.-H. and Nelson, B. L., "A fully sequential procedure for indifference-zone selection in simulation," *ACM Transactions on Modeling and Computer Simulation*, vol. 11, no. 3, pp. 251–273, 2001.

[55] Kim, S.-H. and Nelson, B. L., "On the asymptotic validity of fully sequential selection procedures for steady-state simulation," *Operations Research*, vol. 54, no. 3, pp. 475–488, 2006.

[56] Kim, S.-H. and Nelson, B. L., "Selecting the best system," in *Handbooks in Operations Research and Management Science: Simulation* (Henderson, S. G. and Nelson, B. L., eds.), pp. 501–534, Amsterdam: Elsevier Science, 2006.

[57] Kleywegt, A. J., Shapiro, A., and Homem-de-Mello, T., "The sample average approximation method for stochastic discrete optimization," *SIAM Journal of Optimization*, vol. 12, no. 2, pp. 479–502, 2001.

[58] Kushner, H. J. and Clark, D. S., *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. New York, NY: Springer-Verlag, 1978.

[59] Kushner, H. J. and Yin, G. G., *Stochastic Approximation Algorithms and Applications*. New York, NY: McGraw-Hill, 1997.

[60] L'Ecuyer, P. and Yin, G., "Budget-dependent convergence rate of stochastic approximation," *SIAM Journal of Optimization*, vol. 8, no. 1, pp. 217–247, 1998.

[61] Ljung, L., Pflug, G. C., and Walk, H., *Stochastic Approximation and Optimization of Random Systems*. Basel, Switzerland: Birkhauser Verlag, 1992.

[62] Mak, W.-K., Morton, D. P., and Wood, R. K., "Monte Carlo bounding techniques for determining solution quality in stochastic programs," *Operations Research Letters*, vol. 24, no. 1-2, pp. 47–56, 1999.

[63] Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A., "Convergence and finite time behavior of simulated annealing," *Advances in Applied Probability*, vol. 18, no. 1, pp. 747–771, 1986.

[64] Morito, S., Lee, K. H., Mizoguchi, K., and Awane, H., "Exploration of a minimum tardiness dispatching priority for a flexible manufacturing system–A combined simulation/optimization approach," in *Proceedings of the 1993 Winter Simulation Conference* (Evans, G. W., Mollaghasemi, M., Russell, E. C., and Biles, W. E., eds.), pp. 829–837, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 1993.

[65] Neddermeijer, H. G., van Oortmarssen, G. J., Piersma, N., and Dekker, R., "A framework for response surface methodology for simulation optimization," in *Proceedings of the 2000 Winter Simulation Conference* (Joines, J. A., Barton, R. R., Kang, K., and Fishwick, P. A., eds.), pp. 129–136, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2000.

[66] Nelson, B. L., Swann, J., Goldsman, D., and Song, W., "Simple procedures for selecting the best system when the number of alternatives is large," *Operations Research*, vol. 49, no. 6, pp. 950–963, 2005.

[67] Niederreiter, H., *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: SIAM, 1992.

[68] Ólafsson, S. and Kim, J., "Towards a framework for black-box simulation optimization," in *Proceedings of the 2001 Winter Simulation Conference* (Peters, B. A., Smith, J. S., Medeiros, D. J., and Rohrer, M. W., eds.), pp. 300–306, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2001.

[69] Pflug, G. C., "Sampling derivatives of probabilities," *Computing*, vol. 42, pp. 315–328, 1989.

[70] Pflug, G. C., "On-line optimization of simulated Markovian processes," *Mathematics of Operations Research*, vol. 15, pp. 381–395, 1990.

[71] Pichitlamken, J. and Nelson, B. L., "A combined procedure for optimization via simulation," *ACM Transactions on Modeling and Computer Simulation*, vol. 13, no. 2, pp. 155–179, 2003.

[72] Polyak, B. T. and Juditsky, A. B., "Acceleration of stochastic approximation by averaging," *SIAM Journal on Control and Optimization*, vol. 30, pp. 838–855, 1992.

[73] Prudius, A. A. and Andradóttir, S., "Simulation optimization using balanced explorative and exploitative search," in *Proceedings of the 2004 Winter Simulation Conference* (Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., eds.), pp. 545–549, Institute of Electrical and Electronics Engineeres, Piscataway, NJ, 2004.

[74] Prudius, A. A. and Andradóttir, S., "Two simulated annealing algorithms for noisy objective functions," in *Proceedings of the 2005 Winter Simulation Conference* (Kuhl, M. E., Steiger, N. M., Armstrong, F. B., and Joines, J. A., eds.), pp. 797–802, Institute of Electrical and Electronics Engineeres, Piscataway, NJ, 2005.

[75] Robbins, H. and Monro, S., "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[76] Robinson, S. M., "Analysis of sample path optimization," *Mathematics of Operations Research*, vol. 21, no. 3, pp. 513–528, 1996.

[77] Rubinstein, R. Y. and Kroese, D. P., *The Cross-Entropy Method*. New York, NY: Springer, 2004.

[78] Rubinstein, R. Y. and Shapiro, A., *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. New York, NY: Wiley, 1993.

[79] Ruppert, D., "A Newton-Raphson version of the multivariate Robbins-Monro procedure," *Annals of Statistics*, vol. 13, pp. 236–245, 1985.

[80] Shapiro, A., "Simulation-based optimization–Convergence analysis and statistical inference," *Communications in Statistics–Stochastic Models*, vol. 12, no. 3, pp. 425–454, 1996.

[81] Shapiro, A. and Wardi, Y., "Convergence analysis of stochastic algorithms," *Mathematics of Operations Research*, vol. 21, no. 3, pp. 615–628, 1996.

[82] Shedler, G. S., *Regenerative Stochastic Simulation*. Boston, MA: Academic Press, 1993.

[83] Shi, L. and Ólafsson, S., "Nested partitions method for stochastic optimization," *Methodology and Computing in Applied Probability*, vol. 2, no. 3, pp. 271–291, 2000.

[84] Shi, L. and Ólafsson, S., "Stopping rules for the stochastic nested partitions method," *Methodology and Computing in Applied Probability*, vol. 2, no. 1, pp. 37–58, 2000.

[85] Spall, J. C., *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Hoboken, NJ: Wiley, 2003.

[86] Swisher, J. R., Hyden, P. D., Jacobson, S. H., and Schruben, L. W., "A survey of simulation optimization techniques and procedures," in *Proceedings of the 2000 Winter Simulation Conference* (Joines, J. A., Barton, R. R., Kang, K., and Fishwick, P. A., eds.), pp. 119–128, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2000.

[87] Truong, T. H. and Azadivar, F., "Simulation based optimization for supply chain configuration design," in *Proceedings of the 2003 Winter Simulation Conference* (Chick, S., Sánchez, J., Ferrin, D., and Morrice, D., eds.), pp. 1268–1275, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2003.

[88] Tsitsiklis, J. N., "Markov chains with rare transitions and simulated annealing," *Mathematics of Operations Research*, vol. 14, no. 1, pp. 70–90, 1989.

[89] Vogt, H., "A new method to determine the tool count of a semiconductor factory using FabSim," in *Proceedings of the 2004 Winter Simulation Conference* (Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., eds.), pp. 1925–1929, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2004.

[90] Wieland, F. and Holden, T. C., "Targeting aviation delay through simulation optimization," in *Proceedings of the 2003 Winter Simulation Conference* (Chick, S., Sánchez, J., Ferrin, D., and Morrice, D., eds.), pp. 578–584, Institute of Electrical and Electronics Engineeres, Piscataway, New Jersey, 2003.

[91] Wolpert, D. H. and Macready, W. G., "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[92] Yakowitz, S., "A globally convergent stochastic approximation," *SIAM Journal on Control and Optimization*, vol. 31, no. 1, pp. 30–40, 1993.

[93] Yakowitz, S., L'Ecuyer, P., and Vázquez-Abad, F., "Global stochastic optimization with low-dispersion point sets," *Operations Research*, vol. 48, no. 6, pp. 939–950, 2000.

[94] Yakowitz, S. and Lugosi, E., "Random search in the presence of noise, with application to machine learning," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 4, pp. 702–712, 1990.

[95] Yan, D. and Mukai, H., "Stochastic discrete optimization," *SIAM Journal on Control and Optimization*, vol. 30, no. 3, pp. 594–612, 1992.

VITA

Andrei A. Prudius was born in Sarapul, Russia, on December 11, 1979. He received a B.S. in Industrial Engineering from Boğaziçi University in Istanbul, Turkey, in 2001 and an M.S. in Operations Research from the Georgia Institute of Technology in 2004. His research interests are in simulation, and especially in simulation optimization. After completing his Ph.D. studies in Industrial and Systems Engineering at the Georgia Institute of Technology in 2007, he will be joining the Research and Development Department at Bloomberg L.P. as a quantitative financial developer.